UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

LUCAS EDUARDO AZEVEDO SIMÕES

Técnicas Amostrais para Otimização Não Suave

Sampling Techniques for Nonsmooth Optimization

Campinas

2017

Lucas Eduardo Azevedo Simões

Sampling Techniques for Nonsmooth Optimization

Técnicas Amostrais para Otimização Não Suave

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Matemática Aplicada.

Thesis presented to the Institute of Mathematics, Statistics and Scientific Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Applied Mathematics.

Orientadora: Sandra Augusta Santos

Coorientador: Elias Salomão Helou Neto

Este exemplar corresponde à versão final da Tese defendida pelo aluno Lucas Eduardo Azevedo Simões e orientada pela Profa. Dra. Sandra Augusta Santos.

Campinas

2017

Agência(s) de fomento e nº(s) de processo(s): FAPESP, 2013/14615-7; CAPES
ORCID: http://orcid.org/0000-0002-2305-3565

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Si51s

Simões, Lucas Eduardo Azevedo, 1989-
Sampling techniques for nonsmooth optimization / Lucas Eduardo Azevedo Simões. – Campinas, SP : [s.n.], 2017.

Orientador: Sandra Augusta Santos.
Coorientador: Elias Salomão Helou Neto.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Otimização não diferenciável. 2. Otimização matemática. 3. Otimização irrestrita. 4. Algoritmos. 5. Amostragem (Estatística). I. Santos, Sandra Augusta,1964-. II. Helou Neto, Elias Salomão. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Técnicas amostrais para otimização não suave
Palavras-chave em inglês:
Nonsmooth optimization
Mathematical optimization
Unrestricted optimization
Algorithms
Sampling (Statistics)
Área de concentração: Matemática Aplicada
Titulação: Doutor em Matemática Aplicada
Banca examinadora:
Sandra Augusta Santos [Orientador]
José Mário Martínez Pérez
Lúcio Tunes dos Santos
Claudia Alejandra Sagastizabal
Ademir Alves Ribeiro
Data de defesa: 16-03-2017
Programa de Pós-Graduação: Matemática Aplicada

Tese de Doutorado defendida em 16 de março de 2017 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). SANDRA AUGUSTA SANTOS

Prof(a). Dr(a). JOSÉ MARIO MARTÍNEZ PÉREZ

Prof(a). Dr(a). LÚCIO TUNES DOS SANTOS

Prof(a). Dr(a). CLAUDIA ALEJANDRA SAGASTIZABAL

Prof(a). Dr(a). ADEMIR ALVES RIBEIRO

A Ata da defesa com as respectivas assinaturas dos membros
encontra-se no processo de vida acadêmica do aluno.

Aos meus queridos pais

e

à minha amada esposa...

"A felicidade é salutar para o corpo,
mas só a dor robustece o espírito."
(Marcel Proust)

Agradecimentos

Esta tese possui vários autores, pessoas fundamentais na construção do que aqui é apresentado. Assim, nada mais justo do que demonstrar toda a minha gratidão a estes que trilharam a caminhada ao meu lado.

Agradeço a Deus pelas inspirações diárias, pelas alegrias e dificuldades que experimentei durante estes quatro anos e, sobretudo, por me sustentar em todos os momentos.

Agradeço aos meus pais, Eduardo e Sandra, por acreditarem que o amor e a educação são as melhores heranças que um filho pode ter, me mostrando a beleza da vida e por, pacientemente, me ensinarem como ela deve ser vivida.

Agradeço à minha amada esposa, Francielle, por, tão de perto, acompanhar esta minha jornada, alegrando os meus dias com o seu amor e otimismo e me mostrando que o sol pode brilhar mesmo nos dias mais difíceis.

Agradeço a toda a minha família por sempre me apoiarem e estarem comigo durante todos estes anos. Levo comigo todo o carinho que me deram. Sou um pedaço de cada um de vocês.

Agradeço aos meus orientadores, Sandra e Elias, por sabiamente me guiarem durante todo este período do meu doutorado, por se tornarem mais que meus orientadores, mas meus amigos. Estarei em eterno débito com vocês por tudo que me ensinaram.

Agradeço também a todo o grupo de otimização do IMECC, professores e alunos, pelos apontamentos, sugestões e melhorias durante as apresentações nos nossos seminários semanais, pelo clima amistoso e acolhedor durante todos estes anos.

Agradeço a todos os meus amigos que, para minha imensa felicidade, são muitos. Meu muito obrigado aos que, desde a minha infância, estão presentes na minha vida e aos meus amigos de graduação e pós-graduação que tornaram todos estes dias mais fáceis.

Por fim, meus sinceros agradecimentos à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP - Processo 2013/14615-7), por me apoiarem e tornarem este estudo possível.

Resumo

O método amostral de gradientes (GS) é um algoritmo recentemente desenvolvido para resolver problemas de otimização não suave. Fazendo uso de informações de primeira ordem da função objetivo, este método generaliza o método de máxima descida, um dos clássicos algoritmos para minimização de funções suaves.

Este estudo tem como objetivo desenvolver e explorar diferentes métodos amostrais para a otimização numérica de funções não suaves. Inicialmente, provamos que é possível ter uma convergência global para o método GS na ausência do procedimento chamado "teste de diferenciabilidade". Posteriormente, apresentamos condições que devem ser esperadas para a obtenção de uma taxa de convergência local linear do método GS. Finalmente, um novo método amostral com convergência local superlinear é apresentado, o qual se baseia não somente no cálculo de gradientes, mas também nos valores da função objetivo nos pontos sorteados.

Palavras-chave: otimização não convexa e não suave. minimização irrestrita. métodos amostrais. convergência local. teste de diferenciabilidade.

Abstract

The Gradient Sampling (GS) method is a recently developed tool for solving unconstrained nonsmooth optimization problems. Using just first order information of the objective function, it generalizes the steepest descent method, one of the most classical methods for minimizing a smooth function.

This study aims at developing and exploring different sampling algorithms for the numerical optimization of nonsmooth functions. First, we prove that it is possible to have a global convergence result for the GS method in the abscence of the differentiability check procedure. Second, we prove in which circumstances one can expect the GS method to have a linear convergence rate. Lastly, a new sampling algorithm with superlinear convergence is presented, which rests not only upon the gradient but also on the objective function value at the sampled points.

Keywords: nonsmooth nonconvex optimization. unconstrained minimization. gradient sampling. local convergence. differentiability check.

List of Figures

List of Tables

List of symbols

$\operatorname{co} \mathcal{X}$	convex hull of $\mathcal{X}$;
$\operatorname{cl} \mathcal{X}$	closure of $\mathcal{X}$;
$\operatorname{ri} \mathcal{X}$	relative interior of $\mathcal{X}$;
$\lvert\mathcal{X}\rvert$	cardinality of $\mathcal{X}$;
$\mathcal{B}(x, r)$	Euclidean open ball with center at x and radius r;
$\overline{\mathcal{B}}(x, r)$	Euclidean closed ball with center at x and radius r;
$\lVert \cdot \rVert$	Euclidean norm in $\mathbb{R}^n$ (with exception of Chapter 1 where it stands for any norm in $\mathbb{R}^n$);
$\lVert x \rVert_H$	the positive real value $\sqrt{x^T H x}$, for any symmetric positive definite matrix H;
e	vector of appropriate dimension with ones in all entries;
x_+	a vector where each coordinate is given by $\max\{x_i, 0\}$;
$\operatorname{Proj}_{\mathcal{V}}(x)$	orthogonal projection of x onto the set $\mathcal{V}$;
$\operatorname{Proj}^H_{\mathcal{V}}(x)$	projection of x onto the set $\mathcal{V}$ associated with the norm induced by a positive definite symmetric matrix H;
$A_{i,:}$	the i-th row of the real matrix A;
$A_{:,j}$	the j-th column of the real matrix A;
$\operatorname{Re} \lambda$	the real part of the complex value λ;
$\mathcal{P}[x \in \mathcal{X}]$	probability of x to be in $\mathcal{X}$;
$\mathcal{P}[x \in \mathcal{X} \mid x \in \mathcal{Y}]$	probability of x to be in $\mathcal{X}$ given that $x \in \mathcal{Y}$.

Contents

Introduction

Problems involving continuous nonsmooth functions arise in many fields of science [35, 43, 48], acting in a direct way or playing a secondary role (e.g., subproblems) in different areas. A large class of problems needs to cope with one or more minimizations of convex nonsmooth functions [39, 42], which has been successfully solved by well established optimization algorithms known as *Bundle Methods* [1, 25, 34]. However, a significant amount of problems involve minimizations of nonsmooth functions that are also nonconvex [12, 13], a property that usually introduces an undesirable complexity to the implementation of the aforementioned technique.

Recently, an algorithm known as *Gradient Sampling* (GS) [6, 27] has gained attention for providing good alternatives to the difficulties that Bundle Methods need to deal with if the function is not convex (see [34, 41] and references therein). Basically, the functioning of GS is very close to the steepest descent method for smooth functions, since it works in every iteration with a descent direction computed just with first order information and it finds the next iterate by a line search procedure (in fact, when a nonnormalized version of GS is used to solve a smooth optimization problem, its step asymptotically recovers the direction taken by the steepest descent method). In contrast to the Bundle Method, the GS does not work with a memory of the past iterations, but it tries to gain information about the function by computing gradients of sampled points obtained in each iteration. This behavior is less complex than keeping a history of the last iterations, since in the nonconvex case it is hard to determine whether a past iteration is contributing to construct a good model of the objective function or it is so far from the current iteration that its incorporation to the model might lead to an erroneous local information. As a counterpart, by evaluating the gradients of the sampled points, the GS has a significant cost per iteration.

The present study has the goal to explore the convergence behavior of methods that use sampling techniques to solve unconstrained nonsmooth optimization problems. Although we present some numerical results along the text, this study is essentially theoretical and its contribution can be divided in three parts.

The first major contribution is the preservation of the global convergence result for the GS method when a step called "differentiability check" is not implemented. We present two alternative procedures that replace this undesirable step, which have the advantage of not asking from the user the knowledge of points of nondifferentiability of the objective function [19].

The second contribution is the study of local convergence of the original GS

method in its nonnormalized version [21]. Although the global convergence result is well established, a local rate of convergence has not been stated yet. In the nonsmooth optimization community, there is a belief that in the best case scenario, the GS method would converge linearly and mainly because it can be seen as a generalization of the steepest descent method [10]. However, because of the random nature of GS, this result can not be easily obtained and a mathematical meaning for the "best case scenario" expression is also not trivial.

Finally, we end this thesis by trying to answer a natural question that arises when one studies the GS algorithm. Since the aforementioned algorithm may be viewed as a generalization of the Cauchy method, it is reasonable to wonder whether it would be possible to develop a sampling technique that would generalize the Newton or quasi-Newton method, in the sense that a superlinear convergence result could be achieved. As a result, we present a new sampling algorithm that moves superlinearly to the solution of the nonsmooth optimization problem in some specific iterations of the method [20].

We believe that the results obtained in this text are a step further into the study of a practical algorithm with rapid local convergence to minimize nonsmooth and nonconvex functions (important studies on the matter for nonsmooth and convex functions can be found in [28–30, 37]). The pursuit for such an algorithm has raised efforts of many researchers (an enlightening review can be found in [38]) and up to our knowledge there is no method in the literature that fulfills those features.

1 Theoretical Background

The need to minimize a function $f : \mathbb{R}^n \to \mathbb{R}$ that is not differentiable in its full domain arise in many areas of science [35, 43, 48] and has gained attention from the optimization field along the last four decades, since a large class of problems needs to cope with one or more minimizations of nonsmooth functions [12, 13, 39, 42]. Usually, the problem to be solved can be described by

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t} \quad & x \in \mathcal{X} \subset \mathbb{R}^n. \end{aligned} \tag{1.1}$$

Here, some of the typical hypotheses commonly made over the optimization problems are not assumed. The map f may not be differentiable at every $x \in \mathbb{R}^n$ and, additionally, it might be nonconvex as well. Besides, the set $\mathcal{X}$ is usually represented by means of continuous functions $g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ such that

$$\mathcal{X} := \{x \in \mathbb{R}^n \mid g_i(x) \leqslant 0 \text{ and } h_j(x) = 0,\ \forall i \in \mathcal{I},\ \forall j \in \mathcal{E}\},$$

where $\mathcal{I}, \mathcal{E} \subset \mathbb{N}$ are finite index sets. The functions g_i and h_i may not be necessarily smooth.

A natural example of a nonsmooth problem can be given by the exact penalty function approach for constrained optimization. Under mild assumptions, it is possible to show that there must exist a sufficiently large $\rho > 0$ such that the solution of the unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} \ f(x) + \rho \max\{\|G(x)_+\|_\infty, \|H(x)\|_\infty\},$$

where

$$G(x) = \begin{bmatrix} g_1(x) & \dots & g_{|\mathcal{I}|}(x) \end{bmatrix}^T \quad \text{and} \quad H(x) = \begin{bmatrix} h_1(x) & \dots & h_{|\mathcal{E}|}(x) \end{bmatrix}^T,$$

is equivalent to problem (1.1). Therefore, even for the smooth optimization case, it is a natural approach to see the problem as a nonsmooth minimization one.

Since many constrained optimization problems can be easily transformed into an unconstrained minimization by the penalization approach presented above, from now on, our study will be focused on the problem (1.1) for $\mathcal{X} = \mathbb{R}^n$. So, the case of interest can be stated as

$$\min_{x \in \mathbb{R}^n} f(x). \tag{1.2}$$

Although we are not requiring smoothness of the function f, it is inevitable to impose some structure over the objective function in order to have a convergent algorithm.

An assumption weaker than the differentiability of a function is the hypothesis of Lipschitz continuity. This assumption is important because it still provides desirable properties for the objective function and cuts out some pathological functions. An important implication of this assumption is that, for any unitary $d \in \mathbb{R}^n$ and $t > 0$, we have the finiteness of the quotient

$$\frac{|f(x+td) - f(x)|}{t}, \text{ for all } x \in \mathbb{R}^n.$$

Therefore, from now on, we suppose that f is a local Lipschitz continuous function.

Definition 1.1 (local Lipschitz continuity)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function. We say that f is a locally Lipschitz continuous function if for all $x \in \mathbb{R}^n$, there exist $r > 0$ and $L_r > 0$ such that*

$$|f(z) - f(x)| \leqslant L_r \|z - x\|, \text{ for all } z \in \mathcal{B}(x, r).$$

An important implication of Definition 1.1 is given by Rademacher's theorem [4], which states that if f is locally Lipschitz continuous, then the set of points where the function f is not differentiable has null measure (in the sense of Lebesgue measure). Roughly speaking, one can say that given any point $x \in \mathbb{R}^n$ obtained in a random and uniform way, with probability one, the function f will be differentiable at x. Hence, a point in the domain of the objective function where f is not differentiable is rare.

In addition, we ask a little more structure for the function f. So, defining $\mathcal{D}$ as the set of all points in which f is differentiable, we assume the following property for f along this study.

Assumption 1.1. *The set $\mathcal{D}$ is open in $\mathbb{R}^n$.*

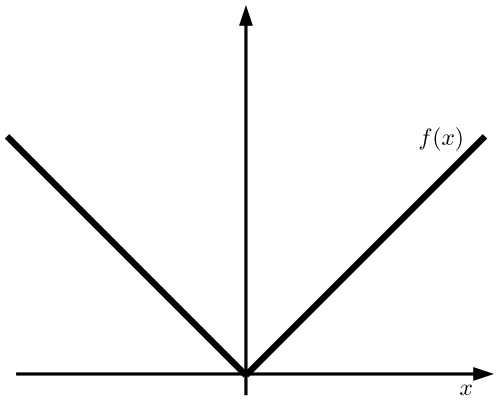


Figure 1 – Absolute value function.

At this point, one could be tempted to think that, in practice, any algorithm developed to minimize smooth functions works to optimize a locally Lipschitz function,

since points of nondifferentiability are rare. Unfortunately, this is not a true statement. Usually, in real-world problems, when the minimization of a locally Lipschitz function is needed, some or all local minimizers are points of nondifferentiability. Consequently, it is easy to see that any method used to solve the optimization will face the nondifferentiability issue. For example, this undesirable behavior can be seen in the one-dimensional absolute value function (see Figure 1) or, more generally, in the norm $\|\cdot\|_1$. Many regularization processes of ill-posed problems use this norm to obtain a good solution and, in general, at the optimal point, the function $\|\cdot\|_1$ is not differentiable. Moreover, some methods might get stuck in regions of nondifferentiability far away from the solution [22, Chapter VIII].

The next section has the goal to establish generalizations of some well known concepts of smooth functions and present useful properties of locally Lipschitz functions.

1.1 Basic concepts

There are several ways of presenting and motivating a generalization of the different concepts related with smooth functions. Additionally, different generalizations are possible. Here, we have decided to use the concepts developed by Clarke [8] and to start giving a geometrical motivation of those notions. For that goal, we begin with a simple definition.

Definition 1.2 (cone)**.** *Given any nonempty set $\mathcal{A} \subset \mathbb{R}^n$, we say that $\mathcal{A}$ is a cone in $\mathbb{R}^n$ if for any scalar $\alpha \geqslant 0$, we have*

$$a \in \mathcal{A} \Rightarrow \alpha a \in \mathcal{A}.$$

Notice that the null vector is always an element of a cone $\mathcal{A}$, since by its definition $0 \cdot a$ must belong to $\mathcal{A}$. With this first concept in mind, we determine an important cone for our study.

Definition 1.3 (tangent cone)**.** *Given any nonempty set $\mathcal{X} \subset \mathbb{R}^n$, we say that $d \in \mathbb{R}^n$ is tangent to $\mathcal{X}$ at $x \in \mathcal{X}$ if for every sequence of points $\{x_j\} \subset \mathcal{X}$ converging to x and any positive sequence $\{t_j\} \subset \mathbb{R}$ decreasing to zero, there is a sequence $\{d_j\} \subset \mathbb{R}^n$ converging to d such that*

$$x_j + t_j d_j \in \mathcal{X}, \quad \text{for all } j \in \mathbb{N}. \tag{1.3}$$

The set that contains all the tangent vectors of $\mathcal{X}$ at x is called tangent cone and is denoted by $T_{\mathcal{X}}(x)$.

Directly linked to the notion of tangent cone is the normal cone. For such a definition, we choose any inner product $\langle \cdot, \cdot \rangle$ in $\mathbb{R}^n$.

Definition 1.4 (normal cone)**.** *Let $\mathcal{X} \subset \mathbb{R}^n$ be a nonempty set and $x \in \mathcal{X}$. Then, the set*

$$N_{\mathcal{X}}(x) := \{v \in \mathbb{R}^n \mid \langle v, d\rangle \leqslant 0, \quad \forall d \in T_{\mathcal{X}}(x)\}$$

is called the normal cone to $\mathcal{X}$ at x.

Therefore, for any nonempty $\mathcal{X} \subset \mathbb{R}^n$, it is not difficult to see that the set $N_{\mathcal{X}}(x)$ is always convex and closed. For our case, the tangent and normal cone will be generally related to the set called epigraph.

Definition 1.5 (epigraph)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be any continuous function. Then, the set*

$$\operatorname{epi} f := \{(x, z) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leqslant z\}$$

is known as the epigraph of f.

Although the epigraph set seems just another representation of the function f, it will be very helpful to motivate and understand the definition of stationary point that will be introduced ahead. For now, let us exhibit two representations of functions with their respective tangent and normal cones (see Figure 2). Notice that there is a significant

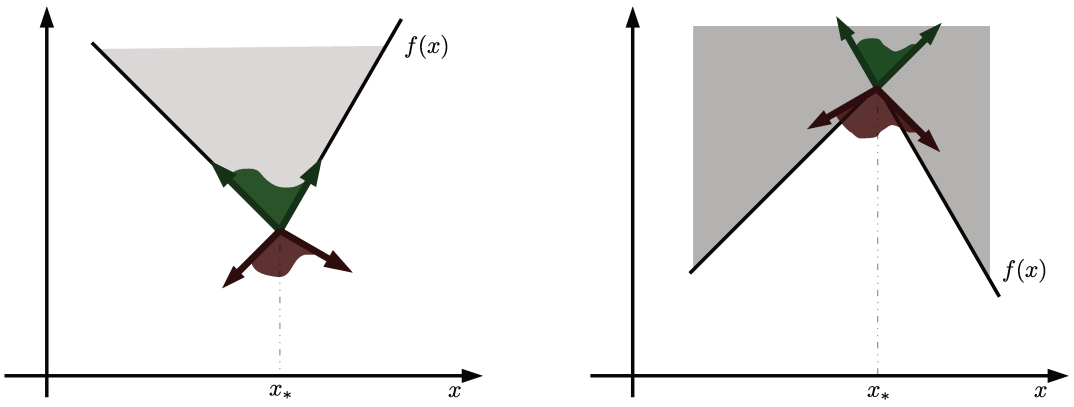


Figure 2 – Two representations of functions. On the left side we have a convex function, which implies a convex epigraph as well (set of points in grey). On the right-hand side we have a nonconvex function with its respective nonconvex epigraph. In both figures, the red and green colors represent the normal cone and the tangent cone, respectively.

difference between the normal cone of a convex and nonconvex set. While in the convex case the normal cone does not "enter" in the set, a nonconvex set can have a normal cone stepping into it.

For motivating this section, we start by giving some physical meaning to the examples shown in Figure 2. Let us suppose that the epigraph of the function represented

on the right-hand side is a solid structure. Moreover, just because we always represent the gravitational field pointing downwards, we flip the picture (see Figure 3). Therefore, let us suppose that a tiny object was left at the position $(x_*, f(x_*))$. Consequently, since the epigraph of the function is solid, the structure will impose a force against the gravity, not allowing the particle to enter into it. However, since this object is in the corner of two "solid walls", there are different manners that the structure could exert this force. Additionally, this force can only be made by the vectors normal to those walls.

Assuming, by simplicity, that $\overrightarrow{g}$ is a normalized vector, that is, $\overrightarrow{g} = (0, -1)$, any vector $(\xi, 1) \in -N_{\operatorname{epi} f}\big(x_*, f(x_*)\big)$, with $\xi \in \mathbb{R}$, is a vector that will exert a force against the gravity (the minus in front of $N_{\operatorname{epi} f}\big(x_*, f(x_*)\big)$ represents the set of vectors with directions opposing those in $N_{\operatorname{epi} f}\big(x_*, f(x_*)\big)$ and it appears in the expression just because we have flipped the original figure). Therefore, any force in the normal cone that exerts a counter force to the gravity is called the generalized gradient (or subgradient) of the nonsmooth function f, since these vectors are the only forces that can impose a movement to the object. Generalizing this concept to the n-dimensional real space, we exhibit our next definition.

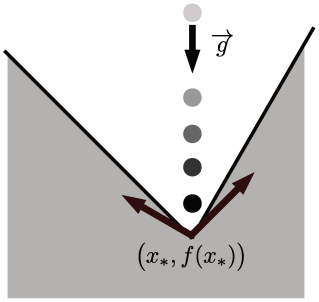


Figure 3 – Representation of a tiny object at the position $(x_*, f(x_*))$ related to right-hand side picture of Figure 2.

Definition 1.6 (subdifferential,subgradient)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function. Then, given any point $x \in \mathbb{R}^n$, we call the set*

$$\overline{\partial} f(x) := \{\xi \in \mathbb{R}^n \mid (\xi, -1) \in N_{\operatorname{epi} f}\big(x, f(x)\big)\}$$

as the subdifferential of f at x. Any $\xi \in \overline{\partial} f(x)$ is called a subgradient of f at x.

Looking at Figure 3, one can see that at the point we are computing the normal cone, it is possible to have a force that will produce an equilibrium over the tiny object, i.e., there is a vector inside the normal cone such that it will not exert any kind of force, but in the opposite direction of the gravity. Indeed, $(0, 1)$ belongs to the set $-N_{\operatorname{epi} f}\big(x_*, f(x_*)\big)$,

or, in a more direct way,

$$(0,-1) \in N_{\operatorname{epi} f}\big(x_*, f(x_*)\big).$$

Hence, since the equilibrium is possible, the object will remain stationary at this point. This reasoning motivates the next definition.

Definition 1.7 (stationary point)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally continuous Lipschitz function. Then, given any fixed point $x \in \mathbb{R}^n$, we call the point x an stationary point for the function f if*

$$(0,-1) \in N_{\operatorname{epi} f}\big(x, f(x)\big),$$

or, alternatively, $0 \in \overline{\partial} f(x)$.

Mathematically, it would be desirable to have that every local minimum (or maximum) of the function f satisfies the definition of stationarity. However, to ensure that this property holds, we need to define the generalized directional derivative for nonsmooth functions.

Notice that we have established a generalization for the derivative concept without using a generalization of the directional derivative. This connection will be given by analyzing the set $T_{\operatorname{epi} f}(x, f(x))$. So, in order to explore the relation between $T_{\operatorname{epi} f}(x, f(x))$ and $\overline{\partial} f(x)$, we will need a well known definition in the convex analysis area.

Definition 1.8 (support function)**.** *Given a compact and convex set $\mathcal{C} \subset \mathbb{R}^n$, we define its respective support function $s : \mathbb{R}^n \to \mathbb{R}$ as*

$$s_{\mathcal{C}}(d) := \max\{\langle x, d\rangle \mid x \in \mathcal{C}\}.$$

The support function is of great importance in convex analysis because a compact and convex set defines a support function and vice-versa, i.e., it is enough to have one of these objects to know both. This result is due to the fact that $\mathcal{C}$ can be described as the intersection of all half-spaces of the form

$$\mathcal{S}^-_{\mathcal{C}}(x) := \{y \in \mathbb{R}^n \mid \langle x, y\rangle \leqslant s_{\mathcal{C}}(x)\}. \quad \text{(see Figure 4)}$$

We will see that the generalized directional derivative will be the support function of the set $\overline{\partial} f(x)$. For now, we present a proposition that will prove useful.

Proposition 1.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function. Then, given $d \in \mathbb{R}^n$, $r \in \mathbb{R}$ and $x \in \mathbb{R}^n$, the following holds*

$$(d, r) \in T_{\operatorname{epi} f}(x, f(x)) \Leftrightarrow \limsup_{\substack{y \to x \\ t \downarrow 0}} \frac{f(y + td) - f(y)}{t} \leqslant r.$$

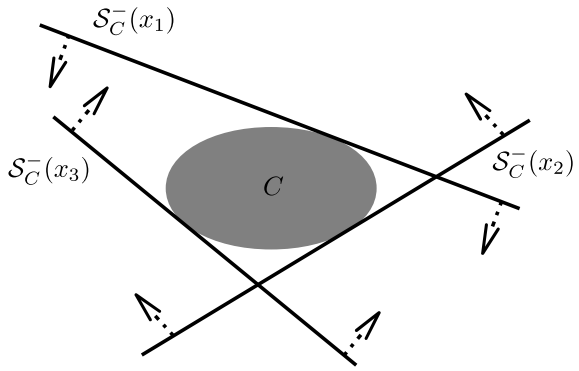


Figure 4 – Representation of a compact and convex set $\mathcal{C}$ and some of its half-spaces $\mathcal{S}_{\mathcal{C}}^-(x_i)$ related to the support function associated with $\mathcal{C}$.

Proof. Given any $(d, r) \in T_{\operatorname{epi} f}(x, f(x))$, we must have, for any sequence $\{(x_j, z_j)\} \subset \operatorname{epi} f$ converging to $((x, f(x))$ and any $\{t_j\}$ decreasing to zero, that there is $\{(d_j, r_j)\} \subset \mathbb{R}^n \times \mathbb{R}$ converging to (d, r) such that

$$(x_j, z_j) + t_j(d_j, r_j) \in \operatorname{epi} f, \text{ for all } j \in \mathbb{N} \Leftrightarrow \frac{f(x_j + t_j d_j) - z_j}{t_j} \leqslant r_j, \text{ for all } j \in \mathbb{N}.$$

Therefore, since this must be valid for any sequence $\{(x_j, z_j)\} \subset \operatorname{epi} f$, we choose the sequence $\{(x_j, f(x_j))\}$, which implies

$$\frac{f(x_j + t_j d_j) - f(x_j)}{t_j} \leqslant r_j, \text{ for all } j \in \mathbb{N} \Rightarrow \limsup_{j\to\infty} \frac{f(x_j + t_j d_j) - f(x_j)}{t_j} \leqslant r. \tag{1.4}$$

Notice that since f is locally Lipschitz continuous, there must exist a Lipschitz constant L that holds for all points near x. Consequently, for j sufficiently large, it yields that

$$\left| \frac{f(x_j + t_j d_j) - f(x_j)}{t_j} - \frac{f(x_j + t_j d) - f(x_j)}{t_j} \right| \leqslant L\|d_j - d\|.$$

Hence, by the condition that $d_j \to d$ and by (1.4), we have

$$\limsup_{j\to\infty} \frac{f(x_j + t_j d) - f(x_j)}{t_j} = \limsup_{j\to\infty} \frac{f(x_j + t_j d_j) - f(x_j)}{t_j} \leqslant r.$$

Summarizing, the following holds

$$(d, r) \in T_{\operatorname{epi} f}(x, f(x)) \Rightarrow \limsup_{\substack{y\to x \\ t\downarrow 0}} \frac{f(y + td) - f(y)}{t} \leqslant r.$$

Conversely, let $(d, r) \in \mathbb{R}^n \times \mathbb{R}$ and assume that

$$\limsup_{\substack{y\to x \\ t\downarrow 0}} \frac{f(y + td) - f(y)}{t} \leqslant r$$

holds. Consequently, for any sequences $\{(x_j, z_j)\} \subset \operatorname{epi} f$ and $\{t_j\} \subset \mathbb{R}$ such that $x_j \to x$ and $t_j \downarrow 0$, it yields that

$$\limsup_{j\to\infty} \frac{f(x_j + t_j d) - z_j}{t_j} \leqslant \limsup_{j\to\infty} \frac{f(x_j + t_j d) - f(x_j)}{t_j} \leqslant r. \tag{1.5}$$

Then, we define the sequence $\{(d, r_j)\} \subset \mathbb{R}^n \times \mathbb{R}$, with

$$r_j = \max\left\{\sup_{s\geqslant j} \frac{f(x_s + t_s d) - z_s}{t_s}, r\right\}.$$

By the manner we have defined the sequence $\{(d, r_j)\}$, we see that

$$\frac{f(x_j + t_j d) - z_j}{t_j} \leqslant r_j, \text{ for all } j \in \mathbb{N} \Rightarrow f(x_j + t_j d) \leqslant z_j + t_j r_j, \text{ for all } j \in \mathbb{N}.$$

Equivalently, we have $(x_j, z_j) + t_j(d, r_j) \in \operatorname{epi} f$. Moreover, by (1.5), it implies that $(d, r_j) \to (d, r)$, which completes the proof of our statement. □

This last result, gives the support function of the closed and convex set $\overline{\partial} f(x)$. Indeed, since $(\xi, -1) \in N_{\operatorname{epi} f}(x, f(x))$, then

$$(\xi, -1) \in N_{\operatorname{epi} f}(x, f(x)) \Leftrightarrow \langle(\xi, -1), (d, r)\rangle \leqslant 0, \text{ for all } (d, r) \in T_{\operatorname{epi} f}(x, f(x)).$$

So, assuming that the inner product $\langle\cdot,\cdot\rangle$ above is the natural inner product for a space originated from a Cartesian product of a space $\mathcal{X}$ and $\mathbb{R}$, that is, $\langle(a, b), (c, d)\rangle = \langle a, c\rangle_{\mathcal{X}} + bd$, then

$$(\xi, -1) \in N_{\operatorname{epi} f}(x, f(x)) \Leftrightarrow \langle\xi, d\rangle \leqslant r, \text{ for all } (d, r) \in T_{\operatorname{epi} f}(x, f(x)).$$

By Proposition 1.1, one can see that

$$\xi \in \overline{\partial} f(x) \Leftrightarrow (\xi, -1) \in N_{\operatorname{epi} f}(x, f(x)) \Leftrightarrow \langle\xi, d\rangle \leqslant \limsup_{\substack{y\to x \\ t\downarrow 0}} \frac{f(y + td) - f(y)}{t}, \tag{1.6}$$

which yields

$$s_{\overline{\partial} f(x)}(d) = \max_{\xi\in\overline{\partial} f(x)} \langle\xi, d\rangle = \limsup_{\substack{y\to x \\ t\downarrow 0}} \frac{f(y + td) - f(y)}{t}. \tag{1.7}$$

That being established, we are ready to understand the concept of the generalized directional derivative. Remember that for the smooth case, the directional derivative at x in the direction d can be defined as the inner product of the gradient at x and the vector d, i.e., the growth rate of the function in the direction d. In our context, we intend to do the same, but since we can have more than one generalized derivative at the same point, we define the generalized directional derivative as the support function that appears in (1.7).

Definition 1.9 (generalized directional derivative)**.** *The generalized directional derivative of a locally Lipschitz continuous function $f : \mathbb{R}^n \to \mathbb{R}$ at x in the direction $v \in \mathbb{R}^n$ is given by*

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

By relation (1.6), it is also possible to define the subdifferential set in a different manner, but equivalent to its previous definition. While Definition 1.6 has a geometric understanding of the generalized derivative, the definition below presents an analytical view.

Definition 1.10 (Subdifferential set,subgradient)**.** *The set given by*

$$\overline{\partial} f(x) := \{\xi \in \mathbb{R}^n | \ \langle \xi, d \rangle \leqslant f^\circ(x; d), \ \ \forall d \in \mathbb{R}^n\}$$

is called the subdifferential set of f at x and any $\xi \in \overline{\partial} f(x)$ is known as a subgradient of f at x.

From this last definition, it is easy to see that when we compute the generalized derivative at a point x, where the function f is differentiable, we obtain exactly the gradient of the function.

Proposition 1.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally continuous Lipschitz function. Then, if at $x \in \mathbb{R}^n$ the function f is differentiable, we have*

$$\overline{\partial} f(x) = \{\nabla f(x)\}.$$

Proof. Let $\xi \in \overline{\partial} f(x)$ be an arbitrary subgradient. Then, it yields that $\langle \xi, d \rangle \leqslant f^\circ(x; d)$, $\forall d \in \mathbb{R}^n$. However, since f is differentiable at x, we have that $f^\circ(x; d) = \langle \nabla f(x), d \rangle$. Then,

$$\langle \xi, d \rangle \leqslant \langle \nabla f(x), d \rangle, \ \text{ for all } d \in \mathbb{R}^n.$$

Taking $d = \xi - \nabla f(x)$, it yields that $\langle \xi - \nabla f(x), \xi - \nabla f(x) \rangle \leqslant 0$. Therefore, $\xi = \nabla f(x)$, which completes the proof. □

The analytical definition of the subdifferential set, gives us the means to prove a result announced before, but we were not yet capable to attest it.

Proposition 1.3. *If $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz continuous function and x is a local minimum or maximum of the function f, then $0 \in \overline{\partial} f(x)$.*

Proof. First, suppose that x is a local minimum of f. Also, we take any sequence $\{t_j\} \subset \mathbb{R}$ decreasing to zero and the constant sequence $\{x_j\}$ with $x_j = x$, for all $j \in \mathbb{N}$. Then, it is clear that for any $j \in \mathbb{N}$ sufficiently large, we have that

$$\frac{f(x_j + t_j d) - f(x_j)}{t_j} \geqslant 0 \Rightarrow f^\circ(x; d) \geqslant 0, \ \text{ for all } d \in \mathbb{R}^n.$$

Therefore, $0 \in \overline{\partial} f(x)$. Now, for the case of local maximum, we need to show that $0 \in \overline{\partial} f(x) \Leftrightarrow 0 \in \overline{\partial}(-f)(x)$. Indeed,

$$\begin{aligned}(-f)^\circ(x;d) &= \limsup_{\substack{y \to x \\ t \downarrow 0}} \frac{(-f)(y+td) - (-f)(y)}{t} \\ &= \limsup_{\substack{z \to x \\ t \downarrow 0}} \frac{f(z-td) - f(z)}{t}, \text{ using } z = y + td \\ &= f^\circ(x;-d).\end{aligned}$$

Consequently,

$$0 \in \overline{\partial} f(x) \Leftrightarrow \forall d \in \mathbb{R}^n, \langle 0, d\rangle \leqslant f^\circ(x;d) \Leftrightarrow \forall d \in \mathbb{R}^n, \langle 0, -d\rangle \leqslant f^\circ(x;-d) \Leftrightarrow 0 \in \overline{\partial}(-f)(x).$$

But a local maximum for f is a local minimum for $(-f)$, therefore it relies on the case already proven, which completes the proof. □

Both definitions of the subdifferential set are important, but they are not so useful in practice. Even an attempt to approximate $\overline{\partial} f(x)$ seems impracticable. Next, we present a different view of the generalized directional derivative that will prove helpful.

Theorem 1.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function. Then, the generalized directional derivative of f at any point $x \in \mathbb{R}^n$ can be written as*

$$f^\circ(x,d) := \limsup_{\substack{y \to x \\ y \in \mathcal{D}}} \langle \nabla f(y), d\rangle,$$

where $\mathcal{D}$ is the set of points in $\mathbb{R}^n$ such that f is differentiable.

Proof. To prove this statement, we first fix any $d \in \mathbb{R}^n$ and take any $\delta > 0$. Moreover, let $\{x_j\} \subset \mathbb{R}^n$ and $\{t_j\} \subset \mathbb{R}$ be any sequences such that $x_j \to x$ and $t_j \downarrow 0$. As a consequence, we define the sequence $\{z_j\} \subset \mathbb{R}^n$, with

$$z_j \in \mathcal{B}\left(x_j, \frac{\delta t_j}{2L}\right) \cap \mathcal{D}, \text{ where } L \text{ is the local Lipschitz constant around } x.$$

Notice that the existence of each z_j is given by the Rademacher's theorem. Now, for all $j \in \mathbb{N}$ sufficiently large, we must have that

$$\frac{f(x_j + t_j d) - f(x_j)}{t_j} \leqslant \frac{f(z_j + t_j d) - f(z_j)}{t_j} + \delta.$$

However, since the sequences $\{x_j\}$ and $\{t_j\}$ are arbitrary, the relation above yields

$$\limsup_{\substack{y \to x \\ t \downarrow 0}} \frac{f(y+td) - f(y)}{t} \leqslant \limsup_{\substack{y \to x \\ y \in \mathcal{D} \\ t \downarrow 0}} \frac{f(y+td) - f(y)}{t} + \delta.$$

Since $\delta > 0$ can be arbitrarily small and the sequences considered in the right-hand side of the last inequality are contained in the sequences in the left-hand side, we have

$$\limsup_{\substack{y\to x\\ t\downarrow 0}} \frac{f(y+td)-f(y)}{t} = \limsup_{\substack{y\to x\\ y\in\mathcal{D}\\ t\downarrow 0}} \frac{f(y+td)-f(y)}{t}.$$

Again, consider arbitrary sequences $\{x_j\} \subset \mathcal{D}$ and $\{t_j\} \subset \mathbb{R}$ such that $x_j \to x$ and $t_j \downarrow 0$. Then, for any $J, I \in \mathbb{N}$ with $J \geqslant I$, it yields that

$$\lim_{J\to\infty} \left\{ \sup_{j\geqslant J} \frac{f(x_j+t_jd)-f(x_j)}{t_j} \right\} \leqslant \lim_{J\to\infty} \left\{ \sup_{\substack{j\geqslant J\\ i\geqslant I}} \frac{f(x_i+t_jd)-f(x_i)}{t_j} \right\} = \sup_{i\geqslant I} \langle \nabla f(x_i), d\rangle.$$

However, since the sequences involved in the inequality above are arbitrary, we have

$$\limsup_{\substack{y\to x\\ t\downarrow 0}} \frac{f(y+td)-f(y)}{t} \leqslant \limsup_{y\to x} \langle \nabla f(y), d\rangle.$$

In fact, this inequality can be proven to be an equality. Indeed, let us choose any $\delta > 0$. Then, for any sequence $\{x_j\} \subset \mathcal{D}$ converging to x, there is a sequence $\{t_j\} \subset \mathbb{R}$ decreasing to zero such that

$$\langle \nabla f(x_j), d\rangle \leqslant \frac{f(x_j+t_jd)-f(x_j)}{t_j} + \delta \Rightarrow \sup_{j\geqslant J} \langle \nabla f(x_j), d\rangle \leqslant \sup_{j\geqslant J} \frac{f(x_j+t_jd)-f(x_j)}{t_j} + \delta.$$

Considering $j \to \infty$ and since $\{x_j\}$ is arbitrary, we get

$$\limsup_{y\to x} \langle \nabla f(y), d\rangle \leqslant \limsup_{\substack{y\to x\\ t\downarrow 0}} \frac{f(y+td)-f(y)}{t} + \delta.$$

Again, since $\delta > 0$ may be arbitrarily small, we finish our proof. □

In finite dimensional spaces, which is our case, the subdifferential set can be seen from a third viewpoint. To present this new definition, we need an auxiliary result [44, Theorem 2.29].

Theorem 1.2 (Carathéodory's theorem)**.** *If x is an element of the convex hull of a non empty set $\mathcal{X} \subset \mathbb{R}^n$, then there exists a maximum of $n+1$ elements of $\mathcal{X}$ such that x is a convex combination of those points.*

We are ready to prove a theorem that will play a key role in the optimization methods that use sampling techniques.

Theorem 1.3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function. Then, the subdifferential set of f at any point $x \in \mathbb{R}^n$ can be written as*

$$\overline{\partial} f(x) := \operatorname{co}\left\{ \lim_{j\to\infty} \nabla f(x_j) \mid x_j \to x, x_j \in \mathcal{D} \right\},$$

where $\mathcal{D}$ is the set of points in $\mathbb{R}^n$ such that f is differentiable.

Proof. To prove the result, we start by showing that the support function of

$$\mathcal{X} := \operatorname{co}\left\{\lim_{j\to\infty} \nabla f(x_j) \mid x_j \to x, x_j \in \mathcal{D}\right\}$$

is the map $d \mapsto \limsup_{\substack{y\to x\\ y\in\mathcal{D}}} \langle \nabla f(y), d\rangle$. In other words, we show that

$$\max_{\xi\in\mathcal{X}} \langle \xi, d\rangle = \limsup_{\substack{y\to x\\ y\in\mathcal{D}}} \langle \nabla f(y), d\rangle, \text{ for all } d \in \mathbb{R}^n.$$

So, let us fix any $d \in \mathbb{R}^n$. Then, there exists $\xi_d \in \mathcal{X}$ such that $\max_{\xi\in\mathcal{X}} \langle \xi, d\rangle = \langle \xi_d, d\rangle$. By the Carathéodory's theorem, we know that there exist sequences $\{x_j^1\},\ldots,\{x_j^{n+1}\}$ converging to x and scalars $\lambda_1,\ldots,\lambda_{n+1} \geqslant 0$ such that

$$\xi_d = \sum_{i=1}^{n+1} \lambda_i \lim_{j\to\infty} \nabla f(x_j^i), \text{ with } \sum_{i=1}^{n+1} \lambda_i = 1.$$

Then,

$$\langle \xi_d, d\rangle = \left\langle \sum_{i=1}^{n+1} \lambda_i \lim_{j\to\infty} \nabla f(x_j^i), d \right\rangle \leqslant \max_{1\leqslant i\leqslant n+1} \left\langle \lim_{j\to\infty} \nabla f(x_j^i), d \right\rangle \leqslant \limsup_{\substack{y\to x\\ y\in\mathcal{D}}} \langle \nabla f(y), d\rangle.$$

As a result, $\max_{\xi\in\mathcal{X}} \langle \xi, d\rangle \leqslant \limsup_{\substack{y\to x\\ y\in\mathcal{D}}} \langle \nabla f(y), d\rangle$, for all $d \in \mathbb{R}^n$. Let us prove the other side of the inequality.

Let us still consider an arbitrary and fixed $d \in \mathbb{R}^n$. By the definition of lim sup, it yields that there exists a sequence $\{x_j\}$ converging to x such that

$$\langle \nabla f(x_j), d\rangle \geqslant \limsup_{\substack{y\to x\\ y\in\mathcal{D}}} \langle \nabla f(y), d\rangle - \frac{1}{j}.$$

However, because of the locally Lipschitz continuity of f, $\|\nabla f(x_j)\|$ is bounded near x. Consequently, there is a subsequence $\{x_{j_k}\}$ converging to x such that $\nabla f(x_{j_k})$ also converges and

$$\left\langle \lim_{j_k\to\infty} \nabla f(x_{j_k}), d \right\rangle \geqslant \limsup_{\substack{y\to x\\ y\in\mathcal{D}}} \langle \nabla f(y), d\rangle.$$

The inclusion,

$$\lim_{j_k\to\infty} \nabla f(x_{j_k}) \in \operatorname{co}\left\{\lim_{j\to\infty} \nabla f(x_j) \mid x_j \to x, x_j \in \mathcal{D}\right\} \tag{1.8}$$

yields that $d \mapsto \limsup_{\substack{y\to x\\ y\in\mathcal{D}}} \langle \nabla f(y), d\rangle$ is the support function of the set that appears in (1.8). But we have showed in Theorem 1.3 that this map is exactly the support function $f^\circ(x;\cdot)$. As any support function defines a compact convex set and vice-versa, the statement follows. □

It is easy to understand the physical concept behind the last statement. Let us return to the example of Figure 3. It seems reasonable that any force that the tiny particle will receive when it is in the corner of the two "rigid walls" will be a convex combination of the normal vectors of each wall. The theorem above says precisely that, since one can view the normal force of one of those walls as $(\nabla f(y), -1) \in N_{\operatorname{epi} f}(y, f(y))$, for any y near x.

Although Theorem 1.3 depends on the collection of all sequences of points of differentiability converging to x, which decreases the practicality of the result, its representation by the actual gradients of the function f is extremely useful. In this fashion, we can think in a natural set to approximate the real subdifferential $\overline{\partial} f(x)$ by computing gradients near x and taking their convex hull. Associated with this idea, we present a more general set than $\overline{\partial} f(x)$.

Definition 1.11 (ϵ-Subdifferential set, ϵ-subgradient, ϵ-stationary point)**.** *The set called ϵ-subdifferential of f at x is given by*

$$\overline{\partial}_\epsilon f(x) := \operatorname{co} \overline{\partial} f(\mathcal{B}(x, \epsilon)).$$

Any $v \in \overline{\partial}_\epsilon f(x)$ is known as an ϵ-subgradient of f at x. Moreover, if $0 \in \overline{\partial}_\epsilon f(x)$, then we say that x is an ϵ-stationary point for f.

This new set includes the information about the function f around the point $x \in \mathbb{R}^n$, towards the implications of Theorem 1.3. Besides, it still depends on the generalized derivatives and not on the gradients. The next set is more in line with a practical approximation of $\overline{\partial} f(x)$:

$$\mathcal{G}_\epsilon(x) := \operatorname{cl} \operatorname{co} \left(\nabla f \left(\mathcal{B}(x, \epsilon) \cap \mathcal{D}\right)\right).$$

This definition is directly linked with $\overline{\partial}_\epsilon f(x)$, because $\overline{\partial}_\delta f(x) \subset \mathcal{G}_\epsilon(x) \subset \overline{\partial}_\epsilon f(x)$, when $\epsilon > \delta > 0$. Moreover, we have that

$$\overline{\partial} f(x) = \bigcap_{\epsilon > 0} \mathcal{G}_\epsilon(x),$$

which evinces the property of $\mathcal{G}_\epsilon(x)$ to approximate the subdifferential set.

So far, we have exhibited the basic concepts associated with locally Lipschitz functions, but we have not looked yet at the optimization problem (1.2). Although the main focus of our study will be the sampling techniques normally used to solve minimization problems, an introduction about bundle algorithms can not be left aside when one studies nonsmooth optimization.

1.2 Bundle techniques

In this section we have the intent to give a brief introduction in one of the most well known algorithms for solving nonsmooth optimization problems. Bundle methods

were originally developed to deal with nonsmooth and convex functions and, unlike the standard subgradient methods, they were built to be descent methods [34, 41]. Posteriorly, nonconvex objective functions were added to the class of maps that bundle methods might handle, but the complexity of these algorithms increases considerably when compared to their convex versions. For the purpose of this section, we will just consider the case where f is a convex and possibly nonsmooth function.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function, then we say that f is convex if its epigraph is a convex set, i.e, given any $\lambda \in [0, 1]$, it yields that

$$x, y \in \operatorname{epi} f \Rightarrow \lambda x + (1 - \lambda)y \in \operatorname{epi} f.$$

Equivalently, we must have that

$$f(\lambda x + (1 - \lambda)y) \leqslant \lambda f(x) + (1 - \lambda)f(y), \text{ for all } x, y \in \mathbb{R}^n.$$

As a consequence, we can establish the main result that bundle methods are based on, which tells us that it is possible to approximate the function f by affine maps that underestimate the values of f.

Proposition 1.4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function. Moreover, assume that f is convex. Then, for all $x, y \in \mathbb{R}^n$ and $\xi \in \overline{\partial} f(x)$, it yields that*

$$f(x) + \langle \xi, y - x \rangle \leqslant f(y).$$

Proof. For any $x, y \in \mathbb{R}^n$, we define $d = y - x$. Consequently, we must have that

$$\begin{aligned}
f^\circ(x; d) &= \limsup_{\substack{z \to x \\ t \downarrow 0}} \frac{f(z + td) - f(z)}{t} \\
&= \limsup_{\substack{z \to x \\ t \downarrow 0}} \frac{f(t(z + d) + (1 - t)z) - f(z)}{t} \\
&\leqslant \limsup_{\substack{z \to x \\ t \downarrow 0}} \frac{tf(z + d) + (1 - t)f(z) - f(z)}{t} \\
&= \limsup_{z \to x} f(z + d) - f(z) \\
&= f(x + d) - f(x) \\
&= f(y) - f(x).
\end{aligned}$$

So, by (1.7), it yields, for all $\xi \in \overline{\partial} f(x)$, that

$$\langle \xi, y - x \rangle \leqslant f^\circ(x; d) \leqslant f(y) - f(x) \Rightarrow f(x) + \langle \xi, y - x \rangle \leqslant f(y),$$

which proves our statement. □

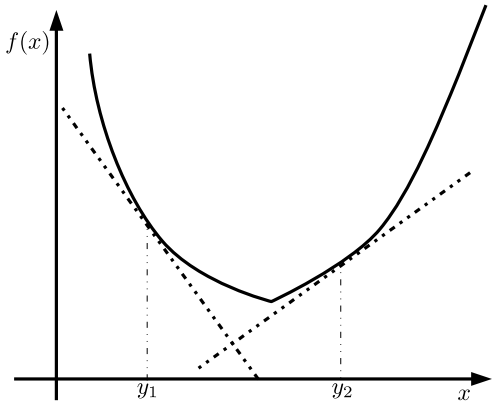


Figure 5 – Representation of a convex function by affine approximations.

The result presented previously is illustrated by Figure 5. In this figure, one can see that the affine approximation in Proposition 1.4 will always underestimate the function f. Moreover, one may consider a model for the objective function using different affine approximations. Basically, the model function can be given by

$$\check{f}(x) := \max_{j \in \mathcal{J}} \{f(y_j) + \xi_j^T (x - y_j)\}, \tag{1.9}$$

where $\mathcal{J} \subset \mathbb{N}$ is a finite set of indices, $y_j \in \mathbb{R}^n$ are points that help us to construct a model for the function f and $\xi_j \in \overline{\partial} f(y_j)$.

The model presented in (1.9) is known as the cutting-plane approach [7, 24] and it is the precursor of bundle methods. Although this model seems a good and cheap way to approximate the actual objective function, it has some drawbacks. For example, the cutting-plane model may not always have a global minimizer, which is a condition closely related to the points $y_j \in \mathbb{R}^n$ that enrich the model of f (see Figure 6). Hence, any attempt to find a good approximation of the solution x_* by looking at $\check{f}$ will prove unsuccessful.

Even in the case that the model $\check{f}$ does present a global minimum, it might be a poor approximation. To illustrate this behavior, let us consider the convex function that appears in Figure 7 with its respective model function $\check{f}$. While the point y_2 is close to the actual solution of the problem, the model that is obtained by using an affine approximation at y_2 gives back a bad approximation of the global minimizer. This happens because at the right-hand side of the solution, the function f is almost horizontal. In that case, if the point y_1 is not close enough to the solution x_*, it will give us a global minimizer of $\check{f}$ far from y_2 and, consequently, from x_*. This explanation also justifies why the cutting-plane method behaves poorly when applied to a smooth function.

Some of the difficulties of the cutting-plane approach exposed here suggest that a stabilization of the method is needed in order to guarantee that $\check{f}$ will always have

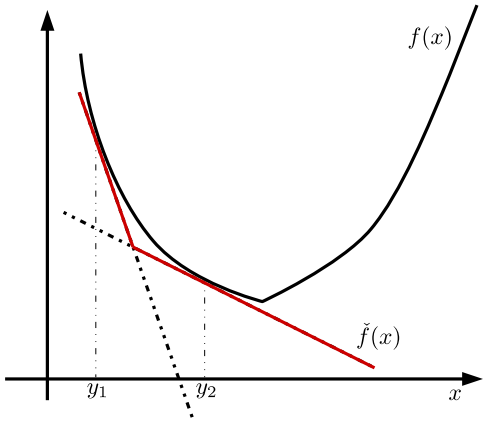


Figure 6 – Representation of a model $\check{f}(x)$ for the function f that does not present a global minimizer.

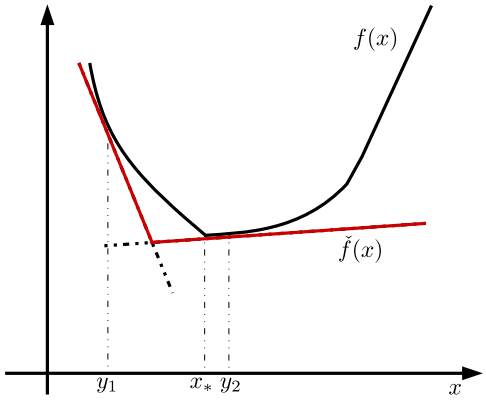


Figure 7 – Representation of how $\check{f}(x)$ can approximate poorly the solution x_*.

a global minimizer and it will not produce bad approximations of x_* when close to the solution. One of the possible ways is to impose a level of reliability on the function $\check{f}$, which means that we will trust in the approximation of $\check{f}$ in a close neighborhood of x_k (the current iteration of the method) [46]. Mathematically speaking, the approximation of the solution x_* will be given by

$$\begin{aligned} \min \quad & \check{f}(x) \\ \text{s.t} \quad & \|x - x_k\| \leqslant \Delta_k, \end{aligned}$$

where $\Delta_k > 0$ controls the radius of reliability of $\check{f}(x)$ around x_k.

Another possible approach that one might consider is to build a quadratic approximation of f instead of considering affine functions. In the case that this quadratic approximation presents a positive definite Hessian, it is clear that a minimum of this approximation will always exist. In view of this goal, it is better to centralize the function

$\check{f}$ at the current iterate point x_k, that is,

$$\check{f}(x) = \max_{j\in\mathcal{J}}\{f(y_j) + \xi_j^T(x - y_j)\} = \max_{j\in\mathcal{J}}\{f(x_k) + \xi_j^T(x - x_k) - \alpha_{k,j}\},$$

where $\alpha_{k,j}$ is called the linearization error and it is given by

$$\alpha_{k,j} := f(x_k) - f(y_j) - \xi_j^T(x_k - y_j).$$

So, considering $H_k \in \mathbb{R}^{n\times n}$ a positive definite matrix, one can stabilize the model of f by giving a new definition for the function $\check{f}$:

$$\check{f}(x_k + d) := \max_{j\in\mathcal{J}}\{f(x_k) + \xi_j^T d - \alpha_{k,j}\} + \frac{1}{2}d^T H_k d.$$

At this moment we are ready to understand the general mechanism of bundle methods. Here, we will discuss a general bundle technique, but we do not have the intent to encompass all the existing bundle methods.

To start, let us consider that we are at an iteration k of our general algorithm with its respective current iterate $x_k \in \mathbb{R}^n$. Moreover, we have an index set $\mathcal{J}_k \subset \{1, \ldots, k\}$ related to the points y_j, with $j \in \mathcal{J}_k$, that enriches our model. As the first step, we solve the following optimization problem for any positive definite matrix $H_k \in \mathbb{R}^{n\times n}$

$$\min_{d\in\mathbb{R}^n} \check{f}(x_k + d) := \min_{d\in\mathbb{R}^n} \max_{j\in\mathcal{J}_k}\{f(x_k) + \xi_j^T d - \alpha_{k,j}\} + \frac{1}{2}d^T H_k d. \tag{1.10}$$

In the way that the above problem is presented to us, it does not seem any easier than our original optimization problem, since the objective function is still nonsmooth. However, the solution of (1.10) can be also obtained by solving the following constrained problem

$$\begin{aligned} \min_{(d,z)\in\mathbb{R}^n\times\mathbb{R}} \quad & z + \frac{1}{2}d^T H_k d \\ \text{s.t} \quad & -\alpha_{k,j} + \xi_j^T d \leqslant z, \;\; j \in \mathcal{J}_k. \end{aligned} \tag{1.11}$$

Now, we are able to obtain a solution by a quadratic optimization problem and there are efficient algorithms in the literature to solve this kind of constrained minimization.

Once solved this subproblem, we define a possible candidate to be the next iterate

$$y_{k+1} := x_k + d_k,$$

with d_k being the solution of (1.11). The acceptance or not of this point as the next iterate will be given by the quotient

$$\frac{f(x_k) - f(y_{k+1})}{f(x_k) - \check{f}(y_{k+1})}.$$

In the case that the quotient above is greater than a positive real number $q \in (0, 1/2)$, we accept the point y_{k+1} as our next iterate, since this relation tell us that the model $\check{f}$ is good

enough (according to the parameter q). Otherwise, a null step is taken, i.e., $x_{k+1} := x_k$. Moreover, we enrich our model by adding the point y_{k+1} and we set $\mathcal{J}_{k+1} := \mathcal{J}_k \cup \{k+1\}$.

The process comes to an end when the predicted decrease of the function value is small enough, i.e., when

$$f(x_k) - \check{f}(y_{k+1}) \leqslant \epsilon,$$

where $\epsilon > 0$ is a previously established tolerance. Finally, we declare the approximate solution of the original optimization problem as the last iterate obtained.

Clearly, different bundle methods can be drawn by setting different ways to choose H_k [25, 26]. Additionally, one can even consider a different quadratic term for each piecewise affine function instead of fixing the same quadratic term for all of them [33].

Independently of the choice of positive definite matrix, an important issue must be stressed at this point. As the number of iterations increases, the index set $\mathcal{J}_k$ will increase. Therefore, the number of restrictions in (1.11) can be dramatically high if a cleaning procedure is not implemented. In the literature, this routine is called aggregation and is an important procedure if one wants to implement a practical method. For this subject, we guide the reader to [2, Chapter 10] and references there in.

As we have already said in the beginning of this section, the bundle techniques are more complex when the hypothesis of convexity is left aside. For this case, it is hard to determine whether a past iteration is contributing to construct a good model of the objective function or it is so far from the current iterate that its incorporation to the model might lead to an erroneous local information. Moreover, the underestimation of the affine approximations is no longer a valid result, which contributes to add an extra difficulty when one tries to build a reliable model for the function f.

With the objective of overcoming these difficulties in solving nonsmooth and nonconvex optimization problems and, in order to present an alternative algorithm to the well established bundle method, the algorithm known as *Gradient Sampling* (GS) was published in 2005 [6]. The next section has the intent to explore its functioning and present the algorithm itself.

1.3 Gradient sampling

In contrast to the bundle techniques, the Gradient Sampling method is not deterministic, since in each iteration a sampling procedure is executed as a way to avoid keeping the past iterations in the memory to construct a good model for f. Consequently, the decision whether the past iteration is contributing to build a good model for f, present in the bundle method, is no longer a concern. On the other hand, the cost per iteration is considerably higher.

In order to understand the GS functioning, we emphasize that the aforementioned method is a generalization of the steepest descent method for nonsmooth functions. Hence, the GS algorithm tries to emulate the behavior of Cauchy method, i.e., in each iteration the algorithm computes a descent direction at the current iterate x_k with just first order information and executes a line search procedure in view of obtaining the next iterate x_{k+1}.

At this point, one can guess that the hardest step is the computation of the descent direction, since the nonsmoothness of the function f can make this procedure harder than when one has a smooth function. Indeed, this step is the kernel of the method and once we understand how this process works, the functioning of the remaining steps becomes natural.

The idea behind the sampling procedure is to approximate the set $\mathcal{G}_\epsilon(x)$ by sampling points around the point x and computing the gradients of f at those points [5]. Of course, the Rademacher's theorem plays a key role at this step, since it states that the set of points in which the function f is not differentiable has null measure. Consequently, if one has, for example, an uniform sampling around x, with probability one the function f will be differentiable at those random points. Therefore, the well-definiteness of the GS method is subjected to an almost sure event.

Additionally, in the case that f is also differentiable at x, the vector $\nabla f(x)$ is also used to approximate $\mathcal{G}_\epsilon(x)$. In further details, considering y_i as the random points sampled in $\mathcal{B}(x,\epsilon)$, we have that

$$\tilde{\mathcal{G}}_\epsilon(x) := \text{co}\{\nabla f(x), \nabla f(y_1), \ldots, \nabla f(y_m)\},$$

is an attempt to approximate $\mathcal{G}_\epsilon(x)$ (for a geometrical view see Figure 8). Moreover, although $\tilde{\mathcal{G}}_\epsilon(x)$ is just an approximation, it has a very important property: independently of the sampled points, it is always possible to find a direction $d \in \tilde{\mathcal{G}}_\epsilon(x)$ of descent for f at x.

Indeed, let us consider that $0 \notin \tilde{\mathcal{G}}_\epsilon(x)$. Since $\tilde{\mathcal{G}}_\epsilon(x)$ is compact, the following vector must exist

$$\tilde{\xi} := \underset{v \in \tilde{\mathcal{G}}_\epsilon(x)}{\arg\min} \langle v, v\rangle. \tag{1.12}$$

We then define, for any $v \in \tilde{\mathcal{G}}_\epsilon(x)$, the function $\phi_v : \mathbb{R} \to \mathbb{R}$, where

$$\phi_v(\lambda) = \left\langle (1-\lambda)\tilde{\xi} + \lambda v, (1-\lambda)\tilde{\xi} + \lambda v \right\rangle.$$

Therefore, it yields that

$$\phi_v'(\lambda) = -2(1-\lambda)\langle \tilde{\xi}, \tilde{\xi}\rangle + 2\lambda\langle v, v\rangle + 2(1-2\lambda)\langle \tilde{\xi}, v\rangle, \text{ for all } \lambda \in \mathbb{R}.$$

However, since $\tilde{\xi}$ is given by (1.12) and $\tilde{\mathcal{G}}_\epsilon(x)$ is convex, we must have that

$$\phi_v(0) \leqslant \phi_v(\lambda), \text{ for all } \lambda \in [0,1],$$

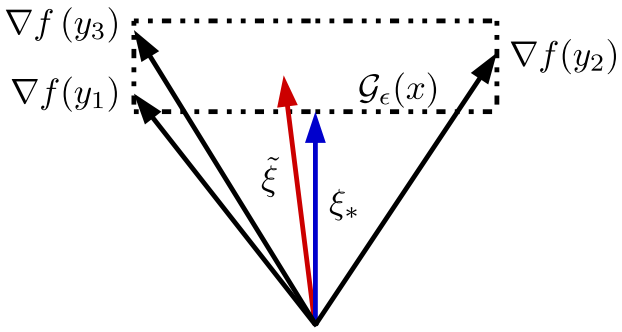


Figure 8 – Representation of the set $\mathcal{G}_\epsilon(x)$ for the bidimensional function $f(x) = \max\{-x_1 + x_2^2, x_1 + x_2^2\}$ at $x = (0, 2)$ with $\epsilon = 0.5$. In this case, we have three sampled points y_1, y_2 and y_3 to approximate $\mathcal{G}_\epsilon(x)$. The vector ξ_* stands for the vector with minimum Euclidean norm over the set $\mathcal{G}_\epsilon(x)$, whereas $\tilde{\xi}$ represents its approximation by just using the gradients $\nabla f(y_1)$, $\nabla f(y_2)$ and $\nabla f(y_3)$.

which implies that ϕ_v is an increasing function at 0. Hence, $\phi_v'(0) \geqslant 0$ and we get

$$-2\langle\tilde{\xi}, \tilde{\xi}\rangle + 2\langle\tilde{\xi}, v\rangle \geqslant 0 \Rightarrow \langle\tilde{\xi}, \tilde{\xi}\rangle \leqslant \langle\tilde{\xi}, v\rangle.$$

Consequently, since $0 \notin \tilde{\mathcal{G}}_\epsilon(x)$ and the above implication is valid for any $v \in \tilde{\mathcal{G}}_\epsilon(x)$, it yields that

$$\langle -\tilde{\xi}, \nabla f(x)\rangle < -\langle\tilde{\xi}, \tilde{\xi}\rangle < 0,$$

which proves that $-\tilde{\xi}$ is a descent direction for f at x.

Having the descent direction $d = -\tilde{\xi}$, there are two possibilities for a given tolerance $\nu > 0$: $\|d\| < \nu$ or $\|d\| \geqslant \nu$. For the first case, the algorithm declares that, under the given tolerance ν, the point x is an ϵ-stationary point for the function f and the method decreases the sampling radius ϵ. Otherwise, a line search procedure is performed along the vector d and finds the next iterate x_{k+1}, which completes an iteration of the method.

There are still some clarifications that we need to provide for a complete understanding of the method, but before we proceed with these explanations, we introduce in Algorithm 1 a general model for most of the GS methods that have been developed over the years [10, 27].

The first observation that we must highlight is the use of the positive definite matrices H_k in the GS algorithm. With those matrices we can define inner products and norms induced by them and, consequently, obtain different concepts of a vector with minimum norm over a closed and convex set. The different types of inner products produce different GS methods. For example, if we set $H_k \equiv I$ for all $k \in \mathbb{N}$, we obtain the standard

GS method. However, updating H_k with limited memory LBFGS techniques, we get a variant of the methods suggested in the study of Curtis and Que [10].

The real value α_k that appears in Step 4 is just an approach to encompass the normalized and nonnormalized versions in the literature. In the pioneering paper of the GS method [6], the original authors make use of a normalized descent direction, while a couple of years later, Kiwiel presented a nonnormalized version of the same idea [27] with better convergence properties. Notice also that the line search procedure in Step 4 is well defined, whenever $x_k \in \mathcal{D}$, since as argued before, d_k will be always a descent direction for f at x_k.

Algorithm 1: A general algorithmic framework for GS methods.

Step 0. Set $k = 0$, $x_0 \in \mathcal{D}$, $m \in \mathbb{N}$ with $m \geqslant n + 1$, fixed real numbers $0 \leqslant \nu_{\text{opt}} < \nu_0$, $0 \leqslant \epsilon_{\text{opt}} < \epsilon_0$ and $0 < \theta_\nu, \theta_\epsilon, \gamma, \beta < 1$.

Step 1. Choose $\{x_{k,1}, \ldots, x_{k,m}\} \subset \mathcal{B}(x_k, \epsilon_k)$ with randomly, independently and uniformly sampled elements.

Step 2. Set $\tilde{G}_k = [\nabla f(x_k)\ \nabla f(x_{k,1})\ \ldots \nabla f(x_{k,m})]$ and find $\tilde{g}_k = H_k^{-1} u_k$ such that $u_k = \tilde{G}_k \lambda_k$ and λ_k solves

$$\begin{aligned} \min_{\lambda} \quad & \frac{1}{2} \lambda^T \tilde{G}_k^T H_k^{-1} \tilde{G}_k \lambda \\ \text{s.t.} \quad & e^T \lambda = 1,\ \lambda \geqslant 0 \end{aligned}$$

where $H_k \in \mathbb{R}^{n \times n}$ is a positive definite symmetric matrix.

Step 3. If $\nu_k < \nu_{\text{opt}}$ and $\epsilon_k < \epsilon_{\text{opt}}$, then STOP!
Otherwise, if $\min\{\|\tilde{g}_k\|, \|\tilde{g}_k\|_{H_k}\} < \nu_k$, then $\epsilon_{k+1} = \theta_\epsilon \epsilon_k$, $\nu_{k+1} = \theta_\nu \nu_k$, $x_{k+1} = x_k$ and go to Step 6.

Step 4. Do a backtracking line search and find the maximum $t_k \in \{1, \gamma, \gamma^2, \ldots\}$ such that

$$f(x_k + t_k d_k) < f(x_k) - \beta \alpha_k t_k \tilde{g}_k^T H_k \tilde{g}_k,$$

where $d_k = -\alpha_k \tilde{g}_k$, for some positive $\alpha_k \in \{1, \vartheta / \|\tilde{g}_k\|\}$. Moreover, set $\epsilon_{k+1} = \epsilon_k$ and $\nu_{k+1} = \nu_k$.

Step 5. If $x_k + t_k d_k \in \mathcal{D}$, then set $x_{k+1} = x_k + t_k d_k$. Otherwise, find $x_{k+1} \in \mathcal{B}(x_k + t_k d_k, \min\{t_k, \epsilon_k\}\|d_k\|) \cap \mathcal{D}$, where the following holds $f(x_{k+1}) \leqslant f(x_k) - \beta \alpha_k t_k \tilde{g}_k^T H_k \tilde{g}_k$.

Step 6. Set $k \leftarrow k + 1$ and go back to Step 1.

Finally, the procedure that has not been discussed yet is the routine that appears in Step 5. It is patent that the necessity of keeping $x_k \in \mathcal{D}$ during all the functioning of the algorithm is a crucial point for the well-definiteness of the method. Without this condition, we cannot compute $\nabla f(x_k)$, compromising the acquisition of a descent direction, and

consequently, collapsing Step 4. Therefore, Step 5 cannot be suppressed without affecting the convergence of GS. Unfortunately, the routine of checking if a point belongs to $\mathcal{D}$ or not is not trivial for a broad class of real problems.

The next chapter is entirely devoted to deal with the difficulty of executing Step 5. Although in most cases the suppression of this procedure does not prevent the method to converge in practice, we show that there are cases where the absence of this step might prevent the achievement of a stationary point for the function f.

2 Differentiability Check

Our purpose in this chapter is threefold. First, we advocate that, even in finite precision, we have to be aware not only about the issues that a point of nondifferentiability might cause, but also about the troubles of a nondifferentiability neighborhood, which opposes to the belief presented so far in many GS variants. Second, we aim to present two modifications in order to produce a gradient sampling algorithm that has probability one to converge, even in the condition where one suppresses the differentiability check (DC). The first proposal considers a perturbation in the search direction of each iteration (if the user does not know the set of nondifferentiability of the function) or just in some specific iterations (if the user does know the set of differentiability or is not tolerant to have too many perturbations). The second one presents a nonmonotone line search as a way to avoid the differentiability check. Third, we exhibit numerical illustrative results and one real problem to show that, besides having a theoretical appeal, our modifications might be useful in practice.

In order to motivate this chapter we present a two dimensional example that illustrates how the GS method might have an undesirable behavior when the DC procedure is not taken into account. Furthermore, we introduce two modifications that have the intent to help the practical algorithm to be well defined and ultimately to have guarantee of convergence.

Let us consider an example with only two variables. Suppose we want to minimize a convex and nonsmooth function $f : \mathbb{R}^2 \to \mathbb{R}$, where

$$f(x) = \max\{\phi_1(x), \phi_2(x), \phi_3(x), \phi_4(x)\}$$

with

$$\begin{aligned} \phi_1(x) &= 0.5x_1^2 + 0.1x_2; \\ \phi_2(x) &= x_1 + 0.1x_2 + 1; \\ \phi_3(x) &= -x_1 + 0.1x_2 + 1; \\ \phi_4(x) &= -0.05x_2 - 50. \end{aligned} \tag{2.1}$$

Assume that we have a point $\overline{x} \in \mathcal{B}((10, 10), 1)$ and we want to start an iteration of the Algorithm 1 with $H_k = I$ and $\alpha_k = 1$ for all $k \in \mathbb{N}$. Therefore, the first step that we need to take is to uniformly sample m points (with $m \geqslant 3$) around $\overline{x}$ with a sampling radius $\epsilon > 0$. For our example, we will set $m = 4$ and $\epsilon = 0.1$. Consider that all of those sampled points were picked up in the set

$$\mathcal{A} := \{(x_1, x_2) \in \mathcal{B}(\overline{x}, \epsilon) \mid x_1 > \overline{x}_1\}. \tag{2.2}$$

This is an event with probability $0.0625 = 0.5^4$ to happen. Thus, since $\phi_1(x)$ is the only function that assumes the maximum in $\mathcal{B}(\overline{x}, \epsilon)$, the result of the minimization of the convex combination that appears in Step 2 is the vector $\nabla\phi_1(\overline{x}) = (\overline{x}_1, 0.1)$. Consequently, assuming that $\nu < 10^{-1}$ in Step 3, the algorithm does not reduce the sampling radius ϵ, and then, the search direction obtained is $d = -\nabla\phi_1(\overline{x})$. Therefore, for any reasonable value of β, it is possible to see that the point

$$x^+ = x + d = (0, \overline{x}_2 - 0.1) \quad \text{(observe that } \overline{x}_2 - 0.1 \geqslant 8.90\text{)}$$

is accepted (assuming that we are suppressing the differentiability check).

Notice that the function f is not differentiable at x^+. Moreover, the algorithm will remain trapped in the manifold

$$\mathcal{M} := \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 = 0\},$$

independently of the sampled points obtained during the next iterations. This behavior is undesirable and troubling for the algorithm, because the function f is not differentiable at any point of

$$\mathcal{W} := \mathcal{M} \cap \{(x_1, x_2) \in \mathbb{R}^2 \mid x_2 \geqslant -340\},$$

and the model algorithm expects that the iterations will always remain in $\mathcal{D}$. However, let us assume that the user returns a reasonable value, namely a subgradient, for the impossible computation of $\nabla f(x)$, for all $x \in \mathcal{W}$. Suppose that the user breaks the 'tie' of the functions arbitrarily by selecting just one function that reaches the maximum to compute the subgradient (which is a recommendation of the original authors [6, Section 4]). Let us say that the user choose to give back the vector $\nabla\phi_2(x)$, for all $x \in \mathcal{W}$ (the same reasoning can be used if the user selects $\nabla\phi_3(x)$).

Now, we must observe that for all iterations in $\mathcal{W}$, the search direction will always be

$$(-1, -0.1) \text{ or } (0, -0.1).$$

The first one is a troubling vector, since it is not a descent direction and will make Step 4 not well defined. However, the probability to sample points that will generate the first vector as a search direction is 6.25%. Moreover, since $x_2^+ \geqslant 8.90$, the algorithm will remain in $\mathcal{W}$ for at least $10 \times (340 + 8.90) = 3489$ iterations. Considering that in 75 iterations the chance to generate the vector $(-1, -0.1)$ as a search direction is $1 - (1 - 6.25/100)^{75} \approx 99\%$, it is possible to see that the line search has an enormous chance of not succeeding at an iteration during the execution of the method.

An observation that one can stress here is that since in practice we do not solve Step 2 with exact precision, the algorithm may never reach the set $\mathcal{W}$ (or this is a very unlikely event), and therefore, this example is not an issue in a real implementation.

Although this statement is partially true, it is false to assume that it will not be problematic for the algorithm (see Figures 9 and 10). Since we are working with finite precision, iterations very close to the set $\mathcal{W}$ may produce the same undesirable behavior as a point that, in fact, is in $\mathcal{W}$. We justify this affirmative exhibiting in Section 2.4 the results of one hundred runs of the nonnormalized version of GS and showing that in 68% of them the algorithm fails to reach the optimal solution.

For now, let us present the modifications that we have proposed for the method. The first one is the perturbation of the search direction.

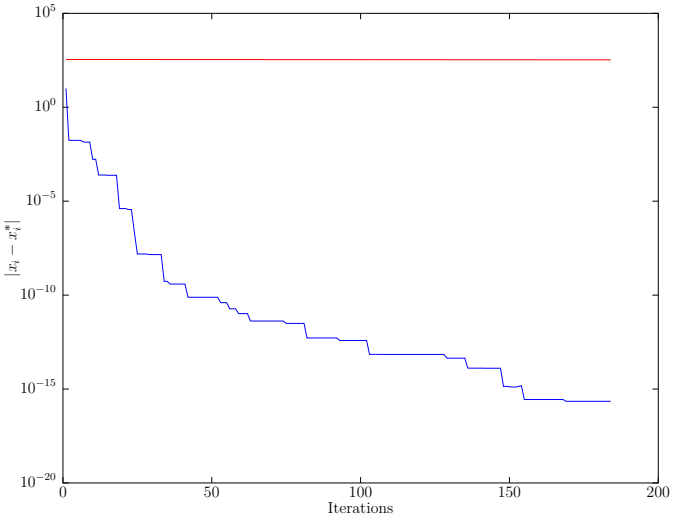


Figure 9 – An example of how the iterations might go quickly to a nondifferentiability region without reaching the optimal solution. In blue color we see the distance of the first coordinate of the iterates to $x_1^* = 0$, whereas the read color stands for the same distance but now for the second coordinate.

2.1 Search direction perturbation

This section has the intent to circumvent the DC procedure by adding a perturbation vector in the search direction, ensuring, with probability one, that every x_k will be in the differentiable set. For this purpose, we suggest that Step 4 of Algorithm 1 is replaced by Step 4a. It is important to notice that if $\nabla f(x_k) = 0$ eventually happens, then Step 4a will never be performed. Indeed, in this case we would have $\tilde{g}_k = 0$, and hence, the algorithm would enter in an infinite loop because of Step 3. Therefore, the quotient that appears inside Step 4a should not worry the reader.

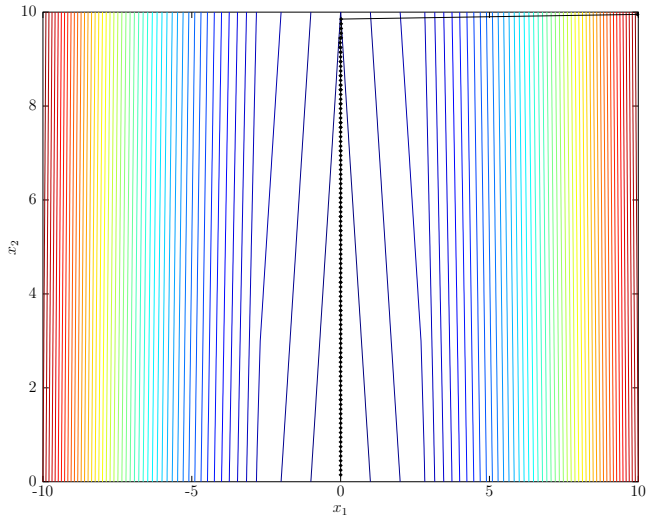


Figure 10 – Here we see a 2D-representation of how the iterations move fast to an undesirable region. The colored lines represent the level curves of the objective function, with the blue colors representing a lower function value than the red lines.

Step 4a. Do a backtracking line search and find the maximum $t_k \in \{1, \gamma, \gamma^2, \ldots\}$ such that

$$f(x_k + t_k d_k) < f(x_k) - \beta \alpha_k t_k \tilde{g}_k^T H_k \tilde{g}_k,$$

where $d_k = -\alpha_k(\tilde{g}_k + \xi_k)$, for some positive $\alpha_k \in \{1, \vartheta/\|\tilde{g}_k\|\}$ and

$$\xi_k \in \mathcal{B}\left(0, c\frac{\nabla f(x_k)^T \tilde{g}_k}{\|\nabla f(x_k)\|}\right)$$

is uniformly and randomly chosen with $0 < c < 1$.

Two important remarks must be stressed here. First, for a minimum disturbance on the natural behavior of GS, one should always set $c \approx 0$. Second, we need to observe that we are trying to avoid the nondifferentiability set whenever this region is far from the solution. Observe that the perturbation vector ξ_k is directly related to the vector $\tilde{g}_k$, whose norm is used as an optimality certificate. Therefore, the perturbation becomes smaller as $\|\tilde{g}_k\|$ goes to zero. So, we are in fact avoiding only the troubling situation in which the algorithm approaches a nondifferentiability neighborhood prematurely.

With this modification, we claim that the differentiability check can be suppressed without affecting the convergence properties. To ensure this, we start proving a lemma.

Lemma 2.1. *If d_k is given by Step 4a, $x_k \in \mathcal{D}$ and x_k is not a stationary point for f,*

then d_k is a descent direction for f at x_k and with probability one $x_k + t_k d_k \in \mathcal{D}$, where t_k is the step size obtained with the Armijo line search.

Proof. First, by relation (4.3) in [10, Lemma 4.3], we know that $\nabla f(x_k)^T \tilde{g}_k \geqslant \|\tilde{g}_k\|^2_{H_k}$. Therefore, it follows that

$$\begin{aligned}
\nabla f(x_k)^T d_k &= -\alpha_k \nabla f(x_k)^T (\tilde{g}_k + \xi_k) \\
&\leqslant -\alpha_k (\nabla f(x_k)^T \tilde{g}_k - \|\nabla f(x_k)\| \|\xi_k\|) \\
&\leqslant -\alpha_k (\nabla f(x_k)^T \tilde{g}_k - c \nabla f(x_k)^T \tilde{g}_k) \\
&= (c-1)\alpha_k \nabla f(x_k)^T \tilde{g}_k \\
&\leqslant (c-1)\alpha_k \|\tilde{g}_k\|^2_{H_k}.
\end{aligned}$$

By assumption, we know that x_k is not stationary for f (so, $\|\tilde{g}_k\|_{H_k} \neq 0$) and $(c-1)\alpha_k < 0$. Consequently, d_k is a descent direction for f at x_k. Now, let us prove that $x_k + t_k d_k \in \mathcal{D}$ with probability one.

First, we define the following isomorphism:

$$\begin{aligned}
T : \mathbb{R}^n &\longrightarrow \mathbb{R}^n \\
x &\longmapsto y = \sigma x + z,
\end{aligned}$$

where $z \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}$ with $\sigma > 0$. Now, given $r > 0$, we define the sets

$$\overline{\mathcal{D}} := \mathcal{D} \cap \mathcal{B}(z, \sigma r) \quad \text{and} \quad \hat{\mathcal{D}} := T^{-1}\left(\overline{\mathcal{D}}\right) \subset \mathcal{B}(0, r).$$

Therefore, considering a uniform distribution and denoting $\mathrm{Vol}(\mathcal{A})$ as the volume of $\mathcal{A} \subset \mathbb{R}^n$, we see that, since $\mathcal{D}$ has full measure in $\mathbb{R}^n$, it implies that

$$\mathrm{Vol}\left(\overline{\mathcal{D}}\right) = \mathrm{Vol}\left(\mathcal{B}(z, \sigma r)\right).$$

Moreover, for any affine transformation $\tilde{T}(x) = Mx + b$, it is well known [49, Section 3.5] that

$$\mathrm{Vol}\left(\tilde{T}(\mathcal{A})\right) = |\det M| \, \mathrm{Vol}\left(\mathcal{A}\right).$$

Therefore, the volume of $\hat{\mathcal{D}}$ exists and is given by

$$\mathrm{Vol}\left(\hat{\mathcal{D}}\right) = \sigma^{-n} \, \mathrm{Vol}\left(\overline{\mathcal{D}}\right).$$

With these facts in mind, we have

$$\begin{aligned}
\mathcal{P}\left[x \in \hat{\mathcal{D}} \mid x \in \mathcal{B}(0, r)\right] &= \frac{\mathrm{Vol}\left(\hat{\mathcal{D}}\right)}{\mathrm{Vol}\left(\mathcal{B}(0, r)\right)} \\
&= \frac{\mathrm{Vol}\left(\overline{\mathcal{D}}\right)}{\mathrm{Vol}\left(\mathcal{B}(z, \sigma r)\right)} \\
&= \mathcal{P}\left[y \in \overline{\mathcal{D}} \mid y \in \mathcal{B}(z, \sigma r)\right] \\
&= 1.
\end{aligned}$$

Consequently, setting $\sigma = -\alpha_k \gamma^j$ and $z = x_k + \sigma \tilde{g}_k$, we see that for a uniform random choice of $\xi_k \in \mathcal{B}(0, r)$, we have, for any $j \in \mathbb{N}$, that $\xi_k \in \hat{\mathcal{D}}$ with probability one, and then,

$$T(\xi_k) = x_k + \sigma \tilde{g}_k + \sigma \xi_k = x_k + \gamma^j d_k \in \overline{\mathcal{D}}$$

with probability one. Therefore, since $t_k \in \{1, \gamma, \gamma^2, \ldots\}$, we have

$$\mathcal{P}[x_k + t_k d_k \in \mathcal{D}] = 1,$$

which completes the proof. □

According to this result, with probability one, the function f is differentiable for all x_k by adding a perturbation vector in the usual direction search. Hence, with probability one, the algorithm is still well defined if we suppress the differentiability check.

Additionally, in order to guarantee the proof of convergence, an assumption over the matrices H_k must be assumed.

Assumption 2.1. *For every $k \in \mathbb{N}$, $H_k \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix and there exist positive real numbers $\underline{\varsigma}$ and $\overline{\varsigma}$ such that*

$$\underline{\varsigma} \|d\|^2 \leqslant d^T H_k d \leqslant \overline{\varsigma} \|d\|^2, \quad \forall d \in \mathbb{R}^n.$$

Now, we can proceed following closely to the results of [27]. We start with a slight modification of [27, Lemma 3.1].

Lemma 2.2. *Suppose that $\mathcal{C} \subset \mathbb{R}^n$ is a nonempty compact and convex set such that $0 \notin \mathcal{C}$. Thus, if $\beta \in (0, 1)$ and $H \in \mathbb{R}^{n \times n}$ is a positive definite symmetric matrix, then there exists $\delta > 0$ such that $u, v \in \mathcal{C}$ and $\|u\|_H \leqslant \operatorname{dist}_H(0, \mathcal{C}) + \delta$ imply $v^T H u > \beta \|u\|_H^2$. Moreover, for a fixed $u \in \mathcal{C}$ and*

$$\mu := \inf_{v \in \mathcal{C}} v^T H u - \beta \|u\|_H^2,$$

it is possible to conclude that $\mu > 0$.

Proof. First, let us prove that there exists $\delta > 0$ such that $u, v \in \mathcal{C}$ and $\|u\|_H \leqslant \operatorname{dist}_H(0, \mathcal{C}) + \delta$ imply $v^T H u > \beta \|u\|_H^2$. Indeed, if the statement were false it would be possible to find sequences $\{u_k\}, \{v_k\} \subset \mathcal{C}$ such that $\|u_k\|_H \leqslant \operatorname{dist}_H(0, \mathcal{C}) + 1/k$ and $u_k^T H v_k \leqslant \beta \|u_k\|_H^2$. Hence, since $\mathcal{C}$ is a compact set, we can assume without loss of generality that $u_k \to \overline{u}$ and $v_k \to \overline{v}$. Thus,

$$\overline{u}^T H \overline{v} \leqslant \beta \|\overline{u}\|_H^2.$$

However, since $\|\cdot\|_H$ is a norm associated with an inner product, we must have by definition of the projection map that

$$\overline{u} = \operatorname{Proj}_{\mathcal{C}}^H(0) \neq 0 \Rightarrow \overline{u}^T H v \geqslant \|\overline{u}\|_H^2, \text{ for all } v \in \mathcal{C},$$

which gives a contradiction.

Now, since $\mathcal{C}$ is a compact set, we have that

$$\inf_{v\in\mathcal{C}} v^T Hu = \min_{v\in\mathcal{C}} v^T Hu.$$

Consequently, there exists $\mu > 0$ such that $\mu = \inf_{v\in\mathcal{C}} v^T Hu - \beta\|u\|_H^2$, which is the desired result. □

Before we present another important result, we expose, for a positive definite symmetric matrix H, some definitions related to the H-measure of proximity to the ϵ-stationarity $\rho_\epsilon^H(\overline{x}) = \text{dist}_H(0, \mathcal{G}_\epsilon(\overline{x}))$:

$$\mathcal{D}_\epsilon^m(x) := \prod_1^m (\mathcal{B}(x,\epsilon)\cap\mathcal{D}) \subset \prod_1^m \mathbb{R}^n$$

and

$$\mathcal{V}_\epsilon^H(\overline{x}, x, \delta) := \left\{(y^1,\dots,y^m)\in\mathcal{D}_\epsilon^m(x) \;:\; \text{dist}_H(0, \text{co}\{\nabla f(y^i)\}_{i=1}^m) \leqslant \rho_\epsilon^H(\overline{x}) + \delta\right\}.$$

We are now able to present our next result, which establishes a lower bound for the step size t_k when the points are properly sampled, and gives us a sufficient condition to ensure that $0 \in \overline{\partial} f(\overline{x})$.

Lemma 2.3. *Let $\epsilon > 0$, $\overline{x}\in\mathbb{R}^n$ and H be a positive definite symmetric matrix.*

i) For any $\delta > 0$, there exist $\tau > 0$ and a nonempty open set $\overline{\mathcal{V}}$ satisfying $\overline{\mathcal{V}} \subset \mathcal{V}_\epsilon^H(\overline{x}, x, \delta)$ for all $x\in\mathcal{B}(\overline{x},\tau)$, that is, $\text{dist}_H\left(0, \text{co}\left\{\nabla f(y^i)\right\}_{i=1}^m\right) \leqslant \rho_\epsilon^H(\overline{x}) + \delta$ for all $(y^1,\dots,y^m)\in\overline{\mathcal{V}}$.

ii) Assuming $0\notin\mathcal{G}_\epsilon(\overline{x})$, pick $\delta > 0$ and $\beta\in(0,1)$ as in Lemma 2.2 for $\mathcal{C} := \mathcal{G}_\epsilon(\overline{x})$, $H = H_k^{-1}$ and then τ and $\overline{\mathcal{V}}$ as in statement (i). Suppose at iteration k of Algorithm 1, Step 5 is reached with $x_k\in\mathcal{B}(\overline{x}, \min\{\tau,\epsilon/3\})$, $\epsilon_k = \epsilon$ and $(x_{k1},\dots,x_{km})\in\overline{\mathcal{V}}$. Then, $u_k\in\mathcal{C}$. Moreover, considering $u = u_k$ in Lemma 2.2 and selecting μ for this fixed u, we have that if $\|\xi_k\| < \mu/L$, where L is the Lipschitz constant over $\mathcal{B}(\overline{x},\epsilon)$, then $t_k \geqslant \min\{1, \gamma\underline{\leq}\epsilon/(6L), \gamma\epsilon/(6\vartheta)\}$.

iii) If $\liminf_k \max\{\|x_k - \overline{x}\|, \|u_k\|, \epsilon_k\} = 0$ with $u_k\in\overline{\partial}_{\epsilon_k} f(x_k)$ for all k, then $0\in\overline{\partial} f(\overline{x})$.

Proof. Let us begin by proving the first statement. Choose $u\in\text{co}\{\nabla f(\mathcal{B}(\overline{x},\epsilon)\cap\mathcal{D})\}$ such that $\|u\|_H < \rho_\epsilon^H(\overline{x}) + \delta$. By Carathéodory's Theorem, we know that there must exist $(x_1,\dots,x_m)\in\mathcal{D}_\epsilon^m(\overline{x})$ such that

$$u = \sum_{i=1}^m \lambda_i \nabla f(x_i), \;\text{ with }\; \sum_{i=1}^m \lambda_i = 1 \text{ and } \lambda \geqslant 0.$$

Since f is continuously differentiable on the open set $\mathcal{D}$, we must have that there exists $\overline{\epsilon} \in (0, \epsilon)$ such that

$$\overline{\mathcal{V}} := \prod_{i=1}^{m} \mathcal{B}(x_i, \overline{\epsilon})$$

is a subset of $\mathcal{D}^m_{\epsilon-\overline{\epsilon}}(\overline{x})$ and

$$\left\| \sum_{i=1}^{m} \lambda_i \nabla f(y_i) \right\|_H < \rho^H_\epsilon(\overline{x}) + \delta, \quad \forall (y_1, \ldots, y_m) \in \overline{\mathcal{V}}.$$

Hence, the statement is proven for $\tau = \overline{\epsilon}$.

To prove the second assertion, we see, by hypothesis, that $(x_{k1}, \ldots, x_{km}) \in \overline{\mathcal{V}} \subset \mathcal{V}^{H_k^{-1}}_\epsilon(\overline{x}, x, \delta)$. Therefore, it follows that $\text{dist}_{H_k^{-1}}(0, \text{co}\{\nabla f(x_{ki})\}_{i=1}^m) \leqslant \rho^{H_k^{-1}}_\epsilon(\overline{x}) + \delta$ and

$$\text{co}\{\nabla f(x_{ki})\}_{i=1}^m \subset \mathcal{G}_\epsilon(\overline{x}).$$

Now, by the manner u_k is computed in Step 2 and as $\nabla f(x_k) \in \mathcal{G}_\epsilon(\overline{x})$ (since $x_k \in \mathcal{B}(\overline{x}, \min\{\tau, \epsilon/3\}) \cap D$) , we have that $u_k \in \mathcal{G}_\epsilon(\overline{x})$ (which also gives us that $\|u_k\| \leqslant L$) and $\|u_k\|_{H_k^{-1}} \leqslant \rho^{H_k^{-1}}_\epsilon(\overline{x}) + \delta$. Hence, by Lemma 2.2, there exists $\mu > 0$ such that

$$\mu = \inf_{v \in \mathcal{G}_\epsilon(\overline{x})} v^T H_k^{-1} u_k - \beta \|u_k\|_{H_k^{-1}}. \tag{2.3}$$

Suppose for contradiction that $t_k < \min\{1, \gamma \underline{\varsigma} \epsilon/(6L), \gamma\epsilon/(6\vartheta)\}$. Hence, the Armijo's inequality does not hold for $\gamma^{-1} t_k$, that is,

$$-\beta\gamma^{-1}\alpha_k t_k \tilde{g}_k^T H_k \tilde{g}_k \leqslant f(x_k + \gamma^{-1} t_k d_k) - f(x_k). \tag{2.4}$$

But we know, from the generalized mean value theorem for Lipschitz functions [8, Theorem 2.3.7], that there exist $y^k \in [x_k + \gamma^{-1} t_k d_k, x_k]$ and $v_k \in \overline{\partial} f(y_k)$ such that

$$f(x_k + \gamma^{-1} t_k d_k) - f(x_k) = \gamma^{-1} t_k v_k^T d_k. \tag{2.5}$$

On the other hand, we observe that

$$\begin{aligned} \gamma^{-1} t_k \|d_k\| &\leqslant \gamma^{-1} t_k \alpha_k (\|\tilde{g}_k\| + \|\xi_k\|) \\ &\leqslant 2\gamma^{-1} t_k \alpha_k \|\tilde{g}_k\|. \end{aligned}$$

Therefore, if $\alpha_k = 1$, it follows that

$$\begin{aligned} \gamma^{-1} t_k \|d_k\| &\leqslant \gamma^{-1} t_k 2 \|\tilde{g}_k\| \\ &\leqslant \gamma^{-1} t_k 2 \|H_k^{-1}\| \|u_k\| \\ &\leqslant \gamma^{-1} t_k 2 \underline{\varsigma}^{-1} L \quad \text{(by Assumption 2.1 and } \|u_k\| \leqslant L) \\ &< \epsilon/3, \end{aligned}$$

otherwise, if $\alpha_k\|\tilde{g}_k\| = \vartheta$, then $\gamma^{-1}t_k\|d_k\| \leqslant \gamma^{-1}t_k 2\vartheta < \epsilon/3$. Thus, since $\|x_k - \overline{x}\| \leqslant \epsilon/3$, we have that $v_k \in \mathcal{G}_\epsilon(\overline{x})$ and also $\|v_k\| \leqslant L$. Now, by (2.4) and (2.5), we have

$$\begin{aligned}
-\beta\gamma^{-1}\alpha_k t_k \tilde{g}_k^T H_k \tilde{g}_k \leqslant \gamma^{-1}t_k v_k^T d_k &\Rightarrow -\beta\alpha_k \tilde{g}_k^T H_k \tilde{g}_k \leqslant v_k^T d_k \\
&\Rightarrow -\beta\alpha_k \tilde{g}_k^T H_k \tilde{g}_k \leqslant -v_k^T \alpha_k(\tilde{g}_k + \xi_k) \\
&\Rightarrow \beta\alpha_k (H_k^{-1}u_k)^T H_k H_k^{-1} u_k \geqslant v_k^T \alpha_k (H_k^{-1}u_k + \xi_k) \\
&\quad \text{(since } \tilde{g}_k = H_k^{-1}u_k\text{)} \\
&\Rightarrow \beta\|u_k\|^2_{H_k^{-1}} \geqslant v_k^T H_k^{-1} u_k + v_k^T \xi_k \\
&\Rightarrow \beta\|u_k\|^2_{H_k^{-1}} \geqslant v_k^T H_k^{-1} u_k - \|v_k\|\|\xi_k\| \\
&\Rightarrow \beta\|u_k\|^2_{H_k^{-1}} > v_k^T H_k^{-1} u_k - \mu \\
&\quad \text{(since } \|v_k\| \leqslant L \text{ and } \|\xi_k\| < \mu/L\text{)} \\
&\Rightarrow \mu > v_k^T H_k^{-1} u_k - \beta\|u_k\|^2_{H_k^{-1}},
\end{aligned}$$

which is a contradiction with (2.3). Therefore, we have the desired lower bound for t_k.

For the last statement, just notice that $\overline{\partial}.f(\cdot)$ is closed, which completes the proof. □

With this result in hands we are ready to prove the convergence of the model algorithm presented in the previous chapter.

Theorem 2.1. *If $\{x_k\}$ is a sequence generated by Algorithm 1 with Step 4a, then either $f(x_k) \to -\infty$ or every cluster point of $\{x_k\}$ is a stationary point for f.*

Proof. By the manner we choose $\{x_{k1}, \ldots, x_{km}\}$, it is possible to see that with probability one the algorithm does not stop in Step 1. Now, we suppose that $\{f(x_k)\}$ has a lower bound $l \in \mathbb{R}$. By the line search inequality, we have that

$$\sum_{k=0}^{\infty} \beta\alpha_k t_k \tilde{g}_k^T H_k \tilde{g}_k \leqslant \sum_{k=0}^{\infty} \left(f(x_k) - f(x_{k+1})\right),$$

and since $f(x_k) \geqslant l$, for all $k \in \mathbb{N}$ and some $l \in \mathbb{R}$, it implies that

$$\sum_{k=1}^{\infty} \alpha_k t_k \tilde{g}_k^T H_k \tilde{g}_k < \infty. \tag{2.6}$$

We also have by Assumption 2.1 that if $\|\tilde{g}_k\| \neq 0$, then

$$\begin{aligned}
\|x_{k+1} - x_k\| &= t_k\|d_k\| \\
&\leqslant t_k\alpha_k(\|\tilde{g}_k\| + \|\xi_k\|) \\
&\leqslant 2t_k\alpha_k\|\tilde{g}_k\| \\
&= 2t_k\frac{\alpha_k}{\|\tilde{g}_k\|}\|\tilde{g}_k\|^2 \\
&\leqslant 2\underline{\varsigma}^{-1}t_k\frac{\alpha_k}{\|\tilde{g}_k\|}\tilde{g}_k^T H_k \tilde{g}_k,
\end{aligned}$$

and therefore, this inequality together with (2.6), give us

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\| \|\tilde{g}_k\| < \infty. \tag{2.7}$$

Now, we split the proof in two cases:

i) $\epsilon_k = \epsilon > 0$ and $\nu_k = \nu > 0$ for all k sufficiently large;

ii) $\epsilon_k, \nu_k \to 0$ and $\{x_k\}$ has a cluster point $\overline{x}$.

In the first case, we have that $\|\tilde{g}_k\|, \tilde{g}_k^T H_k \tilde{g}_k \geqslant \nu$, for all k sufficiently large. By (2.7), the whole sequence must converge, that is, $x_k \to \overline{x}$, for some $\overline{x} \in \mathbb{R}^n$. Moreover, by the locally Lipschitz continuity over a neighborhood of $\overline{x}$, we see that $\|\tilde{g}_k\|$ is bounded, which implies a strictly positive lower bound for α_k. Using this information together with (2.6) and $\tilde{g}_k^T H_k \tilde{g}_k \geqslant \nu$, we have that $t_k \to 0$.

If $0 \notin \mathcal{G}_\epsilon(\overline{x})$ there exist δ, τ, μ and $\overline{\mathcal{V}}$ as in Lemma 2.3. Moreover, since ξ_k is uniformly sampled, there exists, with probability one, an infinite set $\mathcal{K} \subset \mathbb{N}$ such that $\|\xi_k\| < \mu/L$, for all $k \in \mathcal{K}$. Now, since $t_k \to 0$ and $x_k \to \overline{x}$, there exists k_1 such that $t_k < \min\{1, \gamma\underline{\varsigma}\epsilon/(6L), \gamma\epsilon/(6\vartheta)\}$ and $x_k \in \mathcal{B}(\overline{x}, \min\{\tau, \epsilon/3\})$, for all $k \in \mathcal{K}$ and $k \geqslant k_1$. This implies that $(x_{k1}, \ldots, x_{km}) \notin \overline{\mathcal{V}}$ for all $k \in \mathcal{K}$ and $k \geqslant k_1$, which is an event that has probability zero to occur.

On the other hand, if $0 \in \mathcal{G}_\epsilon(\overline{x})$, we can choose $\delta = \nu/2$, for Lemma 2.3 i), and pick $k_2 \in \mathbb{N}$ such that $k \geqslant k_2$ implies that $x_k \in \mathcal{B}(\overline{x}, \tau)$. So, we have that

$$\nu \leqslant \|\tilde{g}_k\|_{H_k} \leqslant \operatorname{dist}_{H_k}\left(0, \operatorname{co}\{\nabla f(x_{ki})\}_{i=1}^m\right), \quad \text{for all } k \geqslant k_2,$$

and consequently, $(x_{k1}, \ldots, x_{km}) \notin \overline{\mathcal{V}}$ for all $k \geqslant k_2$. Again, this is an event that has probability zero to happen. Therefore, with probability one, we must have $\epsilon_k \to 0$.

Consider then that we are in the second case. Then, if $x_k \to \overline{x}$, we have directly from for Lemma 2.3 iii) that $\overline{x}$ is a stationary point. So, let us assume that x_k does to converge to $\overline{x}$. Then, for contradiction, suppose that

$$\liminf_k \max\{\|x_k - \overline{x}\|, \|\tilde{g}_k\|\} \neq 0.$$

Consequently, there must exist $\overline{\nu} > 0$, $\overline{k} \in \mathbb{N}$ and an infinite index set

$$\mathcal{K} := \{k \in \mathbb{N} \mid k \geqslant \overline{k},\ \|x_k - \overline{x}\| \leqslant \overline{\nu}\}$$

such that $\|\tilde{g}_k\| \geqslant \overline{\nu}$, for all $k \in \mathcal{K}$. Hence, by (2.7), it implies that $\sum_k \|x_{k+1} - x_k\| < \infty$. Therefore, $\{x_k\}$ must converge to $\overline{x}$, which is a contradiction. Therefore, by Lemma 2.3 iii), $0 \in \overline{\partial} f(\overline{x})$. □

With the above result we complete the convergence analysis of the model algorithm modified with Step 4a and guarantee a practical procedure to ensure, with probability one, that f will be differentiable at any x_k. Furthermore, we have presented a general convergence proof that embraces several gradient sampling methods, including the algorithms proposed in [6, 10] (with or without the normalization of the search direction).

Observation on the adaptive case. It is important to note that an adaptive approach, in the way Curtis and Que did in [10], can also be introduced in our algorithm without affecting the proofs presented in this section, by just noting that if we infinitely do incomplete line searches during the execution of the algorithm, then the case i) of the proof of Theorem 2.1 can not occur. Indeed, if case i) happens, then we must have that $t_k \to 0$, and since an incomplete line search presents a lower bound for the step size, it is impossible to have $t_k \to 0$. So, we rely on case ii) and with the same proof we see that $\overline{x}$ is a stationary point for f. Otherwise, if at some point of the algorithm, we no longer do incomplete line searches, then from a sufficiently large $k \in \mathbb{N}$ onwards, the algorithm behaves exactly like Algorithm 1, and thus, the same proof holds.

2.2 Alternatives to avoid perturbations in each iteration

For some users, the perturbation procedure in every iteration might sound an excessive precaution. It would be desirable to have a threshold condition to discern when it is really necessary to have a perturbed direction. Although we did not observe any undesirable behavior when perturbing the direction in every iteration (and that is the reason why in our numerical results we always have perturbed the search direction), one might experience an undesirable result.

In this brief section, we have the intent to present two alternatives to avoid the perturbation procedure. The first one is related to the case in which the user does not know the set $\mathcal{D}$ (Case I), whereas the second one occurs if this set can be determined (Case II).

For Case I, one can always avoid the perturbation procedure until a line search has failed, i.e., the step size has become smaller than the machine precision. Indeed, since the perturbation procedure is only executed to guarantee a successful line search, it is reasonable to think that we only need to apply the perturbation when the line search has failed. Therefore, if we have an iteration x_k that does not obtain a successful line search, we can come back to iteration x_{k-1} and find x_k with a perturbed direction, and consequently, we guarantee, with probability one, that x_k will be in $\mathcal{D}$. Moreover, we guarantee that, if x_k is not close enough to a stationary point, x_k will not be in a dangerous neighborhood too close to a nondiferentiability point.

For Case II, if the user observes that the line search gives back an iteration

in which the objective function is not differentiable or is very close to a manifold of nondifferentiability, then the user may reject that point, perturb the search direction and execute the line search again. In that way, there is no need to perform perturbations in every iteration.

Lastly, we want to stress again that we have not encountered any trouble in perturbing the search direction in every iteration. We strongly recommend the precautious user that is concerned with the perturbation procedure to set the parameter c close to zero, prior to any alternative presented in this subsection.

2.3 Nonmonotone line search

This section also focuses on the goal of suppressing the differentiability check without losing the convergence properties of the GS variants. Here, instead of perturbing the search direction, we show that relaxing the line search presented in Step 4 still fulfills our aspirations. What this subsection proposes not only deals with the DC procedure, but also is in agreement with what Kiwiel said in [27, p.385]:

> "Further, the implementation of [6, Section 4] obtained best results for the Armijo parameter $\beta = 0$ (although $\beta > 0$ is required in theory). Thus there is still the need for further study of line searches."

As was shown in Lemma 2.3, a lower bound for the step size t_k exists when the sample points of the current iteration lie in a specific open set. However, we can only assure that it will happen eventually, and therefore, the algorithm could perform many iterations without reaching this specific set. Consequently, due to computer rounding errors we may fail to find a step size t_k greater than zero. Therefore, this modification has the advantage of not only addressing the issue of the differentiability check, but also has the property of avoiding tiny step sizes during the algorithm.

The change that we suggest allows the user not to compute the vector $\nabla f(x_k)$ or replace it by a reasonable vector, that is, a vector that will not make the convex combination of Step 2 to produce a vector that cannot be generated by any convex combination of gradients computed at $\mathcal{B}(x_k, \epsilon_k) \cap \mathcal{D}$. The importance of replacing $\nabla f(x_k)$ by a reasonable vector is essential. For example, if we choose the null vector instead of $\nabla f(x_k)$ the algorithm might erroneously declare that it had reached a stationary point. Thus, we call v_k a *reasonable vector* to replace $\nabla f(x_k)$ if and only if there are $\lambda_1, \ldots, \lambda_p \geqslant 0$, with $\sum_{i=1}^{p} \lambda_i = 1$, and $z_1, \ldots, z_p \in \mathcal{B}(x_k, \epsilon) \cap \mathcal{D}$ such that $\|v_k\| \geqslant \|\sum_{i=1}^{p} \lambda_i \nabla f(z_i)\|$ and v_k is in the cone generated by the vectors $\{\nabla f(z_i)\}_{i=1}^{p}$.

In the sequence, we present the second modification for the GS methods. It is a simple adjustment with a straightforward proof, but it should not belittle the issues that it aims to address.

Step 4b. Do an Armijo's line search and find the maximum $t_k \in \{1, \gamma, \gamma^2, \ldots\}$ such that

$$f(x_k + t_k d_k) < f(x_k) - \beta \alpha_k t_k \tilde{g}_k^T H_k \tilde{g}_k + \eta_k,$$

where $\{\eta_k\}$ is a summable positive sequence and $d_k = -\alpha_k \tilde{g}_k$, for some positive $\alpha_k \in \{1, \vartheta/\|\tilde{g}_k\|\}$.

Since we request that $\sum \eta_k < \infty$, the convergence proof for the Algorithm 1 with Step 4b is essentially the same found in Theorem 2.1.

Theorem 2.2. *Suppose that $\{x_k\}$ is a sequence generated by Algorithm 1 with Step 4b. Furthermore, suppose that in Step 2 we do not compute $\nabla f(x_k)$, but instead, we use $\tilde{G}_k = [\nabla f(x_{k1}) \ldots \nabla f(x_{km})]$ or $\tilde{G}_k = [v_k \ \nabla f(x_{k1}) \ldots \nabla f(x_{km})]$, with v_k being a reasonable vector. Then either $f(x_k) \to -\infty$ or every cluster point of $\{x_k\}$ is a stationary point for f.*

Proof. First, we observe that the only role that the computation of $\nabla f(x_k)$ plays in the entire algorithm is to produce a descent method, and consequently, to have a well defined line search. All the results obtained until this point do not use $\nabla f(x_k)$ besides this reason. Therefore, if one guarantees that the line search will end in a finite number of steps, then not computing the vector $\nabla f(x_k)$ or replacing it by a reasonable vector will guarantee convergence.

We claim that since we have a nonmonotone approach, the line search will always end in a finite number of reductions. Indeed, since f is continuous everywhere, we can find $\bar{t} > 0$ sufficiently small, such that $|f(x_k + td_k) - f(x_k)| < \eta_k/2$, for all $0 < t \leqslant \bar{t}$. Consequently, this statement follows by just noticing that

$$0 < t_k < \min\left\{\bar{t}, \frac{\eta_k}{2\beta\alpha_k \tilde{g}_k^T H_k \tilde{g}_k}\right\}$$

satisfies the inequality presented in Step 4b. Therefore, Step 4b will always be well defined, even if d_k is not a descent direction for f at x_k.

Now, notice that the lower bound for t_k found in Lemma 2.3 *ii*) is still valid, since $\nabla f(x_k)$ is not used to obtain the result and every step size that satisfies the standard line search also satisfies the nonmonotone line search. Therefore, we can follow exactly the same proof of Theorem 2.1 by just noting that the inequalities (2.6) and (2.7) still hold, since $\{\eta_k\}$ is a summable sequence. Moreover, although the reasonable vector may be large, the boundness of $\|g_k\|$ remains valid due to the way g_k is computed. □

We have shown that with this new modification, the GS algorithm does not need to compute $\nabla f(x_k)$ in every iteration, and consequently, there is no longer the need to keep $x_k \in \mathcal{D}$. Moreover, with this nonmonotone line search, the algorithm is much more tolerant with the size of t_k, favoring the method to avoid tiny step sizes.

Remark 2.1. *One can advocate that it is better to proceed like Kiwiel suggested in [27, Procedure 4.3] than to have a nonmonotone line search, since a null step seems to be more reasonable than to have a worse function value. This matter is more delicate than it appears. First, because providing to the algorithm the chance to move, it might allow the method to reach a region in which it is easier to have a successful sample than at the past iteration. Second, because there is a compromise between function evaluations and a new computation of gradients and a solution of a quadratic minimization. Notice that if the number of variables of the problem is large, the number of gradients that the method needs to compute in each iteration can exceed the number of step size reductions that the method might need to reach the machine precision. Therefore, nothing can be stated.*

Remark 2.2. *Although we have presented a method that does not need to use $\nabla f(x_k)$, its use is not forbidden and should be encouraged. In fact, as we have argued, the great difficulty in the practical algorithm is not to be exactly at a point of nondifferentiability (which, indeed, is an unlikely event), but to stay in a close neighborhood of nondifferentiability points. Therefore, the computation of $\nabla f(x_k)$ in a practical algorithm must not be a problem. Consequently, there is no practical reason to avoid its computation and throw away the information therein.*

2.4 Numerical results

In order to see the difficulties that one might face during the algorithm when the DC procedure is ignored in the implementation of GS methods, we have solved illustrative examples to exhibit the bad behavior of these methods in their standard forms. Moreover, we show that our modifications are effective to handle the troubling situation when an iteration is prematurely close to a set in which f is not differentiable. Lastly, we solve a difficult control problem to evince that our changes may also benefit the solution of real problems.

The methods based on gradient sampling employed to obtain the numerical results are: (*i*) the original method (GS) proposed in [6], which uses a normalized search direction; (*ii*) a not normalized version suggested by Kiwiel (nNGS) in [27]; (*iii*) a limited line search version with normalized search direction (LGS) also suggested by Kiwiel and (*iv*) its nonnormalized version (nNLGS). All the tests were implemented using Matlab in an Intel Core 2 Duo T6500, 2.10 GHz and 4 Gb of RAM. We have used `quadprog` as the tool for solving the quadratic minimizations needed in each iteration, setting

`interior-point-convex` as the algorithmic choice and 10^{-12} as the tolerances TolX and TolFun and 10^{-8} (default value) as TolCon. The parameter values used for each of these methods are related in Table 1. Further, if the sampled points at an iteration k were obtained with the radius $\epsilon_k = 10^{-6}$ and Step 2 returns $\|\tilde{g}_k\| < 10^{-6}$, then the algorithm stops and we declare that we have reached our *optimality certificate*.

Algorithm	m	ν_0	ϵ_0	θ_ν	θ_ϵ	γ	β	α_k	H_k
GS	$2n$	10^{-6}	10^{-1}	1	10^{-1}	0.5	0	$\|\tilde{g}_k\|^{-1}$	I
nNGS	$2n$	10^{-6}	10^{-1}	1	10^{-1}	0.5	0	1	I
LGS	$2n$	10^{-6}	10^{-1}	1	10^{-1}	0.5	0	$\|\tilde{g}_k\|^{-1}$	I
nNLGS	$2n$	10^{-6}	10^{-1}	1	10^{-1}	0.5	0	1	I

Table 1 – Parameter values used for the standard implementations of GS.

Both the parameter values and the optimality certificate were chosen to be in agreement with the implementation found in [6]. This is the reason for the choices $\beta = 0$ and $\theta_\nu = 1$. Those values do not agree with the convergence theory, but in [6] the authors realized that, in practice, these settings provide better results. Further, as presented in [6], we also kept safeguards. It was stipulated a maximum number of iterations (10^4 for the illustrative examples and 10^2 for the control problem) per sampling radius. Moreover, for the variants GS and nNGS, if the line search fails during the execution of the methods, we skip the current radius and reduce it to the next one. This last safeguard is important (especially for difficult problems), because a line search failure suggests that the algorithm is not obtaining a good representation of the function f, therefore, it is reasonable to think that a reduction of the sampling radius will provide a better local representation of the same function. Under these observations, we stop the algorithm and declare that it fails to reach the optimality certificate if the algorithm reduces the sampling radius $\epsilon_k = 10^{-6}$ without reaching $\|\tilde{g}_k\| < 10^{-6}$ at this specific ϵ_k (also a procedure implemented by the original authors).

Perturbed and nonmonotone versions of GS. For the perturbed version proposed by this manuscript we have set $c = 10^{-3}$. For the generation of the sequence $\{\eta_k\}$, we have used the Zhang and Hager's nonmonotone line search [50]. Hence, we set $\eta_k = C_k - f(x_k)$, where we define

$$\begin{aligned} \varrho_k \in [\varrho_{\min}, \varrho_{\max}] \ \text{ with } \ 0 \leqslant \varrho_{\min} \leqslant \varrho_{\max} < 1; \\ Q_0 = 1 \ \text{ and } \ Q_{k+1} = \varrho_k Q_k + 1; \\ C_0 = f(x_0) \ \text{ and } \ C_{k+1} = (\varrho_k Q_k C_k + f(x_{k+1}))/Q_{k+1}. \end{aligned} \tag{2.8}$$

Under the hypothesis that $\varrho_{\max} < 1$, it is possible to prove that $\{\eta_k\}$ is a summable and positive sequence (see Appendix A). The value used for ϱ_k is indicated in each problem description.

It is important to mention that we have chosen to use $\nabla f(x_k)$ or a reasonable replacing vector v_k (when necessary) in the nonmonotone approach, instead of using just the gradients of the sampled points. For the illustrative examples, if $x_k \notin \mathcal{D}$, we have broken arbitrarily the 'tie' of the functions by selecting one of them and setting v_k as the gradient of this selected function at x_k. In the control problem, we have assumed that $v_k = \nabla f(x_k)$ is always a reasonable vector. Finally, to indicate our versions of the gradient sampling methods we added the prefixes P (if we have used Step 4a) and nM (if we have used Step 4b) in each method name.

Limited line search variants (LGS and nNLGS). For the variants that use limited line search, we have chosen to implement the following procedure instead of Step 4 and to use $\nabla f(x_k)$ or a reasonable vector when needed at Step 2.

Limited Line Search. Do a limited Armijo's line search and find the maximum $t_k \in \{1, \gamma, \gamma^2, \ldots, \gamma^{l_k}\}$ such that

$$f(x_k + t_k d_k) < f(x_k) - \beta \alpha_k t_k \tilde{g}_k^T H_k \tilde{g}_k,$$

where $d_k = -\alpha_k \tilde{g}_k$, for some positive $\alpha_k \in \{1, \vartheta/\|\tilde{g}_k\|\}$, and l_k is the largest positive integer such that

$$l_k \leqslant -\log_{\gamma^{-1}} \left(\min \left\{ 1, \frac{\gamma \epsilon_k}{3\|d_k\|} \right\} \right).$$

If such t_k does not exist, then go to Step 5.

This is not the only way to implement the Procedure 4.3 contained in [27], however, this choice seemed reasonable for us when we take into account the way the other variants were implemented. Further, by the same reasoning used in [27], it is possible to see that this line search still provide a convergence result, even if we add a perturbation in the search direction or use a nonmonotone line search.

Is is important to mention that by no means we had the ambition to conclude in this study that our modifications are better than the variants LGS and nNLGS. The results that are presented here for the methods LGS and nNLGS, and their perturbed and nonmonotone versions, are just for completeness and to guarantee that the reader has as much information as possible. Again, we stress that the main goal of this study is to show that the differentiability check cannot be ignored if we want that the most used variants of GS be well defined and convergent. Further, we were concerned to present solutions that would keep the new algorithm as close as possible to the original GS.

2.4.1 Motivating examples

In this subsection we exhibit the numerical results when one tries to solve the two dimensional motivating example that appears at the beginning of this chapter and a simpler version of the same problem.

All the initial points were randomly chosen in $\mathcal{B}((10,10),1)$ and for the non-monotone version of the GS variants we have used $\varrho_k = 10^{-1}$, for all $k \in \mathbb{N}$. We minimized the following two dimensional real-valued functions

$$f_{\text{mot}}(x) = \max_{1 \leqslant i \leqslant 4}\{\phi_i(x)\} \text{ and } f_{\text{smot}}(x) = \max_{2 \leqslant i \leqslant 4}\{\phi_i(x)\},$$

where the functions ϕ_i, for $i \in \{1,\ldots,4\}$, are given in (2.1). Both functions have their minimizers at $(x_1^*, x_2^*) = (0,-340)$ with optimal values $f_{\text{mot}}(x^*) = f_{\text{smot}}(x^*) = -33$. Since the GS methods have a non deterministic step, we need to solve each function several times in order to have statistical relevance of the results. Thus, we minimized each function one hundred times for each variant. The results are found in Tables 2 and 3. We named f_m as the median of the last function value of all runs, T_m as the median time to solve all successful runs and $\#Eval$ as the median of function evaluations of the runs that the method demanded to solve the problem.

To determine if the variant has solved the problem, we took the best result $\overline{f}$ obtained by all runs of all variants and we stipulated that a GS method has minimized the problem successfully if the function value of the last iteration is smaller than $\overline{f} + 10^{-4}$.

Observing the results obtained by the minimization of both functions we can see that our changes have provided a significant improvement in the robustness of the methods. This becomes clearer if we look at the values of f_m for each variant. Since the iterations prematurely start to be very close to a region of nondifferentiability for f, the GS and nNGS methods start to have line search failures due to rounding errors. Consequently, the implemented algorithms stop without reaching the optimality certificate and the methods fail to obtain a satisfactory solution to the problems. It is worth mentioning that this undesirable behavior has occurred asking only for a simple decrease of the function at each iteration, i.e. $\beta = 0$. The results can be even worse if one sets $\beta > 0$.

Finally, we observe that the simpler version of the motivating example can be transformed into a linear minimization problem, which evidences the simple structure of that function. To support the statement that even in very simple problems the implemented GS methods might fail, we present in the next subsection a naive minimization problem and some generalizations such an example.

Algorithm	# successful runs	β	f_m	T_m	$\#Eval$
GS	25	0	-19.79312	3.63	1550.00
nNGS	32	0	-0.27127	33.45	15448.00
nM-GS	100	10^{-8}	-33.00000	4.04	1301.00
nM-nNGS	100	10^{-8}	-33.00000	33.33	9073.50
P-GS	100	10^{-8}	-33.00000	3.64	1348.00
P-nNGS	100	10^{-8}	-33.00000	33.63	12738.50
LGS	100	0	-33.00000	4.40	1557.00
nNLGS	100	0	-33.00000	43.91	14643.50
nM-LGS	100	10^{-8}	-33.00000	4.45	1618.50
nM-nNLGS	100	10^{-8}	-33.00000	44.25	14412.00
P-LGS	100	10^{-8}	-33.00000	4.39	1528.50
P-nNLGS	100	10^{-8}	-33.00000	44.93	14793.00

Table 2 – Minimization results for the motivating example.

Algorithm	# successful runs	β	f_m	T_m	$\#Eval$
GS	22	0	-19.77259	3.60	1518.00
nNGS	36	0	-0.51889	33.30	13725.00
nM-GS	100	10^{-8}	-33.00000	3.99	1299.00
nM-nNGS	100	10^{-8}	-33.00000	33.06	9115.50
P-GS	100	10^{-8}	-33.00000	3.62	1334.50
P-nNGS	100	10^{-8}	-33.00000	33.29	12692.50
LGS	100	0	-33.00000	4.47	1537.00
nNLGS	100	0	-33.00000	42.00	13404.50
nM-LGS	100	10^{-8}	-33.00000	4.52	1636.00
nM-nNLGS	100	10^{-8}	-33.00000	42.19	13364.50
P-LGS	100	10^{-8}	-33.00000	4.52	1601.00
P-nNLGS	100	10^{-8}	-33.00000	42.07	13015.00

Table 3 – Minimization results for the simpler version of the motivating example.

2.4.2 Naive example and its generalizations

Here, we show a simple example to endorse that we do not need to contrive an elaborated function to face problematic runs of GS methods. Further, two more general functions were created to provide problems with different dimensions.

Again, one hundred runs were performed for each GS variant and we have set $\varrho_k = 10^{-1}$ for all nonmonotone versions. All the starting points were randomly chosen in $\mathcal{B}(0, 1)$.

The first function that was minimized is a two dimensional function described by $f_{\text{naive}}(x) = 100|x_1| + |x_2 - 500|$. It is easy to see that its minimizer is $(x_1^*, x_2^*) = (0, 500)$ and the optimal value is $f_{\text{naive}}(x^*) = 0$. This minimization could be converted into a linear problem too, but it is even simpler than the examples presented so far. The results are shown in Table 4.

Algorithm	# successful runs	β	f_m	T_m	$\#Eval$
GS	32	0	74.06361	5.04	3427.50
nNGS	0	0	332.23060	–	–
nM-GS	100	10^{-8}	0.00000	5.07	1704.00
nM-nNGS	100	10^{-8}	0.00000	5.13	2274.13
P-GS	100	10^{-8}	0.00000	4.97	2037.50
P-nNGS	100	10^{-8}	0.00000	5.05	2533.50
LGS	100	0	0.00001	6.47	2272.50
nNLGS	100	0	0.00001	5.90	3060.50
nM-LGS	100	10^{-8}	0.00000	6.51	2435.00
nM-nNLGS	100	10^{-8}	0.00001	6.14	3329.00
P-LGS	100	10^{-8}	0.00000	6.45	2316.50
P-nNLGS	100	10^{-8}	0.00001	6.00	3115.00

Table 4 – Minimization results for the function f_{naive}.

Since the iterations quickly go to a region too close to $\mathcal{M} := \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 = 0\}$, any iteration with a bad set of sample points can generate a search direction that will break the line search down. Hence, in most cases, the algorithm stops prematurely, and as a consequence, the methods do not find the desirable solution (see f_m). On the other hand, it is possible to see that our proposed modifications can handle the dangerous neighborhood in a satisfactory way, and hence, the modified algorithms were able to find good solutions for the minimization problem.

Taking advantage of the structure of this example, it is possible to create a general function that, in many cases, will produce the same difficulties for GS methods. We can see f_{naive} as a sum of two separable functions: $f_1(x_1) = 100|x_1|$ and $f_2(x_2) = |x_2 - 500|$. Therefore, when one tries to minimize f_{naive}, the algorithm is solving two distinct and independent minimization problems. The major obstacle for GS algorithms is that the iterations approach very quickly to the minimizer of f_1, forcing the method to work for a long time in (or too close to) a nondifferentiability set. With this reasoning and for n multiple of 4, we have created a function

$$\begin{aligned} g_{\text{split}} : \mathbb{R}^{3n/4} \times \mathbb{R}^{n/4} &\longrightarrow \mathbb{R} \\ (x, y) &\longmapsto g_1(x) + g_2(y), \end{aligned}$$

where

$$g_1(x) = 100 \left(\max_{1 \leqslant i \leqslant 3n/4+1} \{A_{i,:}x\} \right) \quad \text{and} \quad g_2(y) = \|y - 500e\|_1.$$

The matrix $A \in \mathbb{R}^{(3n/4+1)\times(3n/4)}$ can be any full-rank matrix such that each element is between -1 and 1 and satisfies

$$A_{p,:} = -\sum_{i=1}^{3n/4} A_{i,:}, \quad \text{where } p = 3n/4 + 1. \tag{2.9}$$

Algorithm	# successful runs	β	f_m	T_m	#Eval
GS	0	0	1216.94309	–	–
nNGS	0	0	985.43729	–	–
nM-GS	100	10^{-8}	0.00001	120.39	30675.50
nM-nNGS	100	10^{-8}	0.00001	73.23	21665.50
P-GS	100	10^{-8}	0.00001	198.68	91200.00
P-nNGS	100	10^{-8}	0.00001	118.46	57305.50
LGS	100	0	0.00001	260.04	51079.50
nNLGS	100	0	0.00001	146.43	42169.00
nM-LGS	100	10^{-8}	0.00001	257.71	51158.00
nM-nNLGS	100	10^{-8}	0.00001	138.87	39316.50
P-LGS	100	10^{-8}	0.00001	291.36	57828.50
P-nNLGS	100	10^{-8}	0.00001	148.25	42061.00

Table 5 – Minimization results for the function g_{split} for $n = 12$.

Notice that since A has full rank and (2.9) holds, it yields that the unique minimizer of g_1 is $x^* = 0$. Moreover, it is straightforward to see that g_2 assumes its minimum value at $y^* = 500e$.

The function g_{split} has the same features of f_{naive}, and consequently, a minimization algorithm will converge fast to the minimizer of g_1, providing the same undesirable behavior to GS methods. To support this statement, we show the results obtained for $n = 12$ in Table 5.

It is undeniable the difficulty that GS and nNGS methods present when one tries to minimize g_{split}. In none of the runs the algorithms were able to successfully solve the problems. In contrast, our versions have solved 100% of the runs. The same behavior was observed with $n = 4$ and $n = 8$.

Now, just to show that this behavior of the standard algorithms does not appear only for separable functions, we have created a second generalization function of the naive example. Let us consider that we have a number of variables n multiple of 4 and a function

$$\begin{aligned} g_{\text{nsplit}} : \mathbb{R}^{n/4} \times \mathbb{R}^{n/4} \times \mathbb{R}^{n/2} &\longrightarrow \mathbb{R} \\ (x, y, z) &\longmapsto \tilde{g}_1(x, z) + \tilde{g}_2(x, y), \end{aligned}$$

where

$$\tilde{g}_1(x, z) = 100 \left(\max_{1 \leqslant i \leqslant 3n/4+1} \left\{ A_{i,:} \begin{pmatrix} x \\ z \end{pmatrix} \right\} \right) \text{ and } \tilde{g}_2(x, y) = \|x\|^2 + \|y - 500e\|_1,$$

and A is any matrix with the same structure defined for the function g_{split}. Clearly, this new map is not separable, but still, the GS and nNGS are not able to solve the minimization problem. For the results obtained with $n = 12$, see Table 6.

Algorithm	# successful runs	β	f_m	T_m	#Eval
GS	0	0	1158.47939	–	–
nNGS	0	0	889.88880	–	–
nM-GS	100	10^{-8}	0.00001	126.92	30355.00
nM-nNGS	100	10^{-8}	0.00001	78.96	21866.00
P-GS	100	10^{-8}	0.00001	217.24	93824.00
P-nNGS	100	10^{-8}	0.00001	126.42	57353.50
LGS	100	0	0.00001	297.34	55478.50
nNLGS	100	0	0.00001	146.69	39115.00
nM-LGS	100	10^{-8}	0.00001	316.56	58812.50
nM-nNLGS	100	10^{-8}	0.00001	143.46	38045.00
P-LGS	100	10^{-8}	0.00001	313.32	58738.00
P-nNLGS	100	10^{-8}	0.00001	141.00	37900.50

Table 6 – Minimization results for the function g_{nsplit} for $n = 12$.

To conclude our numerical results we exhibit in the next subsection a non trivial real problem in order to reinforce that our procedures have a practical appeal.

2.4.3 Stability problem of a Boeing 767

This problem was presented and for the first time solved in [6]. It is a real problem that comes from the design optimization of a controller of an airplane (Boeing 767) at flutter condition. This maximization problem is far from being trivial and it has the difficulty of having badly scaled data. This characteristic makes the optimization process very difficult and even a tiny improvement on the function value is a tough task when we are close to a local optimizer. To a better understanding of the optimization problem involved in this situation, we must describe some stability measures of a dynamical system $\dot{x} = Ux$, where U is a square matrix. However, we refer the reader to look at [6] to have a more complete introduction on the matter.

One of the ways to measure the stability of a dynamical system is by means of a parameter known as spectral abscissa, which is defined as

$$\alpha(U) = \max\{\text{Re } \lambda \mid \lambda \text{ is an eigenvalue of } U\}.$$

Thus, it is said that the dynamical system $\dot{x} = Ux$ is stable if we have $\alpha(U) < 0$. A more efficient measure is a function known as the distance to instability, defined as

$$d_{\text{inst}}(U) = \min\{\delta \in \mathbb{R}_+ \mid \|U - X\| \leqslant \delta \text{ and } X \text{ is an unstable matrix}\}.$$

The higher is the value of this function, the more stable is the dynamical system. Therefore, the aim of this problem is to maximize the distance to instability of the matrix

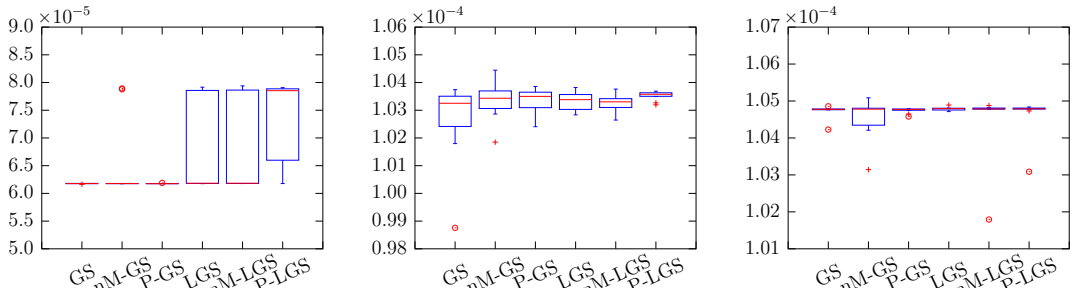
$$M = \begin{bmatrix} A & 0 \\ 0 & 0_k \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & I_k \end{bmatrix} \begin{bmatrix} X_1 & X_2 \\ X_3 & X_4 \end{bmatrix} \begin{bmatrix} C & 0 \\ 0 & I_k \end{bmatrix},$$

Algorithm	β	$\overline{f}$	f_m	T_m	#Eval
GS	0	6.18098e-05	6.17690e-05	609.22	2295.00
nM-GS	10^{-8}	7.89284e-05	6.17696e-05	856.83	2793.50
P-GS	10^{-8}	6.18867e-05	6.17646e-05	683.72	2263.00
LGS	0	7.91686e-05	6.18287e-05	1259.71	5674.50
nM-LGS	10^{-8}	7.93924e-05	6.18313e-05	1287.60	5878.50
P-LGS	10^{-8}	7.90832e-05	7.85363e-05	1863.44	6569.50

Table 7 – Maximization results for the stability problem of an airplane using $k = 0$.

where $A \in \mathbb{R}^{55\times55}$, $B \in \mathbb{R}^{55\times2}$ and $C \in \mathbb{R}^{2\times55}$ are fixed matrices and $X_1 \in \mathbb{R}^{2\times2}$, $X_2 \in \mathbb{R}^{2\times k}$, $X_3 \in \mathbb{R}^{k\times2}$ and $X_4 \in \mathbb{R}^{k\times k}$ are variable matrices[1]. We have considered three instances of this problem ($k \in \{0, 1, 2\}$) with the respective dimensions: $n \in \{4, 9, 16\}$.

For this optimization problem we have chosen to use the algorithms GS, nM-GS, P-GS, LGS, nM-LGS and P-LGS. Since the function that we are trying to maximize is badly scaled, it is reasonable to think that a normalization on the search direction at each iteration is preferable than not normalizing it. Moreover, for the same reason, we used a smaller perturbation parameter ($c = 10^{-6}$) for the P-GS and P-LGS methods, since a small perturbation might lead to a large variation in the function value. Therefore, we kept our attention only on the aforementioned six methods.



(a) $k = 0$ (b) $k = 1$ (c) $k = 2$

Figure 11 – Boxplots of the objective function values obtained with ten runs of each instance.

In order to produce the initial points x_0 that were taken for each instance of the problem, we have used perturbations proportional to the solution obtained by the nM-GS method for the spectral abscissa minimization problem. As the problem demands a considerable computational time, we solved each instance ten times (as the original authors have proceed too). The results can be found in Figure 11 and Tables 7, 8 and 9. Here, we have named $\overline{f}$ as the best function value obtained, f_m as the median of the last function value, T_m as the median time and $\#Eval$ as the median of function evaluations of all runs.

[1] The data of this problem can be found in www.cs.nyu.edu/overton/papers/gradsamp/probs.

Algorithm	β	$\overline{f}$	f_m	T_m	#Eval
GS	0	1.03743e-04	1.03252e-04	1847.89	3744.50
nM-GS	10^{-8}	1.04446e-04	1.03434e-04	2059.54	3859.50
P-GS	10^{-8}	1.03857e-04	1.03501e-04	1927.94	3749.50
LGS	0	1.03821e-04	1.03382e-04	2407.12	7595.50
nM-LGS	10^{-8}	1.03765e-04	1.03303e-04	2369.41	7565.00
P-LGS	10^{-8}	1.03688e-04	1.03576e-04	2402.80	7603.00

Table 8 – Maximization results for the stability problem of an airplane using $k = 1$.

Algorithm	β	$\overline{f}$	f_m	T_m	#Eval
GS	0	1.04856e-04	1.04776e-04	3686.07	3913.00
nM-GS	10^{-8}	1.05087e-04	1.04779e-04	4338.22	5010.00
P-GS	10^{-8}	1.04791e-04	1.04772e-04	4083.01	5151.50
LGS	0	1.04891e-04	1.04792e-04	4781.40	7650.50
nM-LGS	10^{-8}	1.04879e-04	1.04783e-04	4898.98	7534.50
P-LGS	10^{-8}	1.04839e-04	1.04788e-04	4990.90	7737.00

Table 9 – Maximization results for the stability problem of an airplane using $k = 2$.

Observing the results it is possible to see that in none of the instances the standard GS method was able to obtain the best function value. We see that for $k = 0$ the best result is obtained by nM-LGS, whereas in the other instances the algorithm nM-GS has reached the best value. The reason for this positive results against GS is that the other approaches enabled the algorithm to work with fewer line search failures, which allowed the method to reach a better solution. Furthermore, we were able to obtain better function values even with $\beta > 0$. This was prohibitive in the original method, since it would led to many line search failures, preventing the achievement of significant results.

2.4.4 Differences between nM-GS, P-GS and LGS

Although this study does not have the intent to compare our versions with LGS, it is worth stressing the differences of each version, since this clarification might help a future user of gradient sampling methods. To that goal, we exhibit a three dimensional example.

Suppose we have a separable function $f : \mathbb{R}^3 \to \mathbb{R}$ such that $f(x, y, z) = 100f_1(x, y) + f_2(z)$, with

$$f_1(x, y) = \max\{2x + \cot(\theta)y, -2x + \cot(\theta)y, -\cot(\theta)y\},\ f_2(z) = |z|,$$

and $0 < \theta < \pi/2$. Notice that f has the same features of the other functions presented so far, i.e., a minimization algorithm finds quickly the minimizer of f_1, forcing it to work in (or very close to) a nondifferentiability region. Now, let us suppose that we have an iteration (x_k, y_k, z_k) such that $(x_k, y_k) = 0$ (since we have rounding errors, assuming $(x_k, y_k) = 0$ or

very close to 0 makes no difference in the practical computation) and $z_k > 0$ (the same reasoning can be used for $z_k < 0$).

With some simple computations, it is possible to see that, for any sampling radius $\epsilon_k > 0$, the probability to sample a point in

$$\tilde{X} := \{(x, y, z) \in \mathbb{R}^3 \mid -\cot(\theta)y \text{ assumes the maximum in } f_1\}$$

is $(\pi - 2\theta)/(2\pi)$, or in a short way,

$$\mathcal{P}[s \in \tilde{X} \mid s \in \mathcal{B}((x_k, y_k, z_k), \epsilon_k)] = \frac{\pi - 2\theta}{2\pi}.$$

Therefore, we see that if $\theta \to \pi/2$, the probability to have a sampled point in $\tilde{X}$ tends to zero (see Figure 12). To our aim here, let us assume that $\theta = 1.5$, and hence, the probability to sample a point in $\tilde{X}$ is approximately 2.3%. Consequently, if we use $m = 2 \cdot 3$, the probability for none of the sampled points to be in $\tilde{X}$ is approximately 87%, and therefore, there is a high probability to obtain $d_k = -(0, 100\cot(\theta), 1)$ as the search direction.

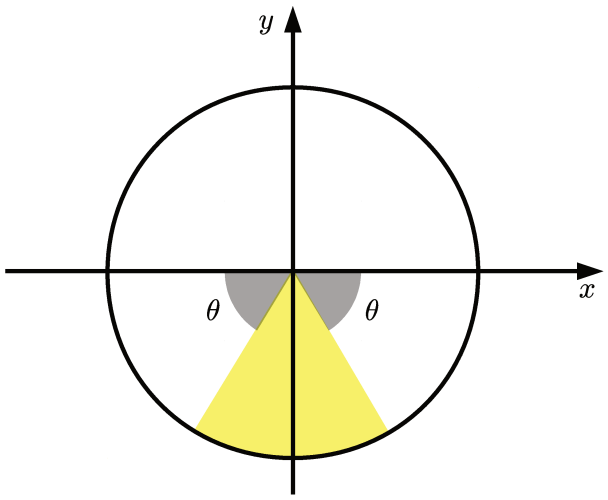


Figure 12 – Representation of the region $\tilde{X}$ (yellow color).

Supposing an iteration of the nNLGS and considering that

$$d_k = -(0, 100\cot(\theta), 1)$$

is not a descent direction for f at (x_k, y_k, z_k), we see that nNLGS has a high chance to accept null steps many times before it allows the algorithm to move. On the other hand, if we consider the nonmonotone version of GS in the situation that the algorithm did not sample in $\tilde{X}$, it will allow a move to another point of the domain. Moreover, the algorithm will move to the specific region that is hard to sample, i.e., to $\tilde{X}$. In that case, $x_{k+1} \in \tilde{X}$,

and therefore, if we use $\nabla f(x_{k+1})$ in the quadratic minimization of Step 2 (like we have used in our illustrative tests), we will no longer need a sampled point in $\tilde{X}$ to have the best search direction. Thus, giving to the algorithm the chance to move, we can land in a region in which it is easier to obtain a good sample and as a consequence, we will move faster than we would if we keep accepting null steps.

For the perturbed version of GS, it will avoid the nondifferentiability region, and therefore, it will accept positive steps in each iteration, allowing the algorithm to move, and again, it might reach a region in the domain that is easier to sample.

Finally, we want to stress that this is the very same behavior that we observe in Tables 5 and 6. For example, considering the random matrix that we have generated in g_{split}, the chance to sample a point such that $A_{2,:}x$ assumes the maximum in g_1 is approximately 2.23%. As a consequence, using $m = 2 \cdot 12$, the chance that in an iteration we do not sample in a region where $A_{2,:}x$ reaches the maximum is approximately 58.20%. Therefore, once the LGS or nNLGS methods reach the nondifferentiability region of g_1, they have more than fifty percent of chance to accept a null step. Hence, more than a half of the computational effort is useless. That is the reason why the proposed methods have such good results when compared to LGS or nNLGS.

2.5 Discussion

In this chapter we have presented a model algorithm for the well known class of gradient sampling methods and pursued ways to overcome an important drawback of these algorithms. The differentiability check has always been a theoretical trick to guarantee the convergence of the methods, but none of the practical algorithms currently existing had this step implemented in their routines, as such kind of verification is impossible for general problems. Moreover, it was argued that this disregard is harmless for a practical implementation, since the event $x_k \notin \mathcal{D}$ is unlikely to occur in practice.

In opposite direction, we have shown illustrative examples where both in theory and in practice the GS fails to reach an acceptable solution if one ignores the DC procedure. This undesired behavior can be explained by the finite precision of the machine. In fact, $x_k \notin \mathcal{D}$ is an unlikely event, however, due to rounding errors, being too close to a set where f is not differentiable might cause the same difficulties as if $x_k \notin \mathcal{D}$. This fact produces an extra difficulty for the user, since one needs to be concerned not only with the differentiability of f at x_k, but also if x_k is not too close to a dangerous set of points.

We have presented two ways to avoid this issue. The first one works with a perturbation vector in the search direction of each iteration. It was shown that, with probability one, all the iterates x_k remain in the differentiable set, and consequently, such a test (differentiability check) can be suppressed without affecting the convergence proof.

The second proposal is a nonmonotone line search. This modification supports the assessment of the differentiability, because even in the unlikely event that $x_k \notin \mathcal{D}$ or x_k is close to a nondifferentiability set for f, the positive term η_k in the backtracking inequality will allow a successful line search. Additionally, it highlights the perennial issue of tiny steps during the execution of these methods, specially in solving some challenging problems. This difficulty shows up due to the nature of these methods, which are random by design. To guarantee the step size t_k remains bounded away from zero, we must sample points in a certain open set. Unfortunately, although we can assure that this event will happen, we cannot know how many iterations it would take for this to happen. As a result, the algorithms may need tiny step sizes to satisfy the backtracking inequality, and due to rounding errors, this eventually generates null steps. Consequently, the algorithms make efforts to produce tiny (or even none) improvements.

In order to show that our changes may also be useful to address real problems, we have solved a challenging control problem. Due to line search failures the GS method stops the runs prematurely and since nM-GS, LGS and nM-LGS are more tolerant, they were able to find better solutions.

Finally, since our proposals are very cheap in terms of computational time and can be easily implemented, we believe that users worried about robustness may be favored by our modifications. Further, the limited line search algorithm presented by Kiwiel (LGS) or a hybrid method (for example, nM-LGS) might be good alternatives as well.

3 Local Convergence Analysis of GS

In 2007, Kiwiel introduced a nonnormalized version of GS [27], which can be seen as a generalization of the well known steepest descent method. Hence, it suggests that, in the best case scenario, the Gradient Sampling will have the same local convergence as the Cauchy method. Although this is reasonable to expect, to the best of our knowledge, there is no proof in the literature of local convergence rates for the GS method nor a clarification of hypotheses under which this can be established.

This chapter has the goal to prove that, under special circumstances, one can achieve linear reduction of the function values at infinitely many iterations of the GS method. Moreover, we justify our hypotheses with illustrative examples, which help us understand when such a behavior cannot be expected.

Looking into the GS functioning, it is possible to see that the method only uses points at which the objective function is differentiable. Additionally, [27, Theorem 3.3] guarantees, with probability one, that $\{x_{k,1}, \ldots, x_{k,m}\} \subset \mathcal{D}$, for all $k \in \mathbb{N}$. This behavior releases the user from returning a subgradient to the method, but, in contrast, the efficiency of the algorithm depends on how well the sampled points describe properly the local behavior of f. Consequently, any local convergence result will be restricted to a good set of sampled points as well.

The next section establishes reasonable conditions over the set of sampled points in order to achieve a linear rate of convergence for the GS method. However, a more structured optimization problem must be assumed. Henceforward, we suppose that x_* is a local minimizer of the objective function f and there is a neighborhood $\mathcal{W} \subset \mathbb{R}^n$ of x_* such that

$$f(x) = \max_{1 \leqslant i \leqslant r} \{\phi_i(x)\}, \quad \text{for all } x \in \mathcal{W}. \tag{3.1}$$

where $r \in \mathbb{N}$ and the functions $\phi_i : \mathbb{R}^n \to \mathbb{R}$ are all of class C^2. It is important to stress that the functions ϕ_i are not analytically known, i.e., we do not assume that the functions ϕ_i are inputs for the GS method. In other words, the representation (3.1) of the function f only plays its role within the theoretical proofs presented in this study. Therefore, we just assume that the method needs to know how to evaluate f and its gradient, whenever the latter exists.

Finally, for functions satisfying (3.1), it is possible to define some sets that will be useful latter. The first one is called the active set of indices at $x \in \mathcal{W}$ and it is given by

$$\mathcal{I}(x) := \{i \in \{1, \ldots, r\} \mid f(x) = \phi_i(x)\},$$

whereas the other sets are defined below (see [36] for more details about U,V-spaces).

Definition 3.1 (U,V-spaces)**.** *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ satisfies (3.1) and x is any point in $\mathcal{W}$. Then, we define*

$$U(x) := \{s \in \mathbb{R}^n \mid [\nabla\phi_i(x) - \nabla\phi_j(x)]^T s = 0, \ \ \forall i, j \in \mathcal{I}(x), \ \ i \neq j\}$$

and $V(x) := U(x)^\perp$ as, respectively, the smooth and nonsmooth subspaces of f at x.

The subspaces defined above are of great importance to us. Notice that they split the domain of the function in two subspaces: the one in which f behaves smoothly (U-space) and the other that captures all the nonsmoothness of the function (V-space).

3.1 Example

With the aid of motivational examples, we settle, in this section, some hypotheses over the set of sampled points to ensure a good descent direction at a given iteration k. Since we do not ask for the knowledge of the functions that comprise f, it seems natural that a good reduction on the function value can only be ensured when the sampled points carry indirect information about the behavior of the functions ϕ_i. This concept is described in the following assumption, where we consider $x_{k,0} := x_k$.

Assumption 3.1. *The sequence $\{x_k\}$ generated by Algorithm 1 converges to the local minimizer x_* and $\epsilon_k, \nu_k \to 0$ according to the global convergence theorem of the GS method. In addition, there is an infinite index set $\mathcal{K} \subset \mathbb{N}$ such that for all $k \in \mathcal{K}$ and any $i \in \mathcal{I}(x_*)$, there is $x_{k,j} \in \mathcal{B}(x_k, \epsilon_k) \subset \mathcal{W}$, for some $j \in \{0, \ldots, m\}$, such that*

$$\phi_i(x_{k,j}) > \phi_s(x_{k,j}), \ \ \textit{for any } s \in \{1, \ldots, r\}\backslash\{i\}. \tag{H_ϕ}$$

At first, H_ϕ seems to demand too much of the GS method. However, as we will see latter (see Remark 3.1), this condition can be naturally obtained (with probability one) from other more common assumptions made in nonsmooth analysis. For now, let us highlight with two examples the importance of H_ϕ for a good local behavior of the GS method.

Let us consider a bidimensional function $f : \mathbb{R}^2 \to \mathbb{R}$, with

$$f(x) = \max\{\phi_1(x), \phi_2(x), \phi_3(x)\},$$

where, for $x = (\xi_1, \xi_2)$, we have

$$\phi_1(x) = \xi_1 + \xi_2, \ \phi_2(x) = -2\xi_1 + \xi_2 \text{ and } \phi_3(x) = \xi_1 - 2\xi_2.$$

Clearly, it is a convex function with $x_* = 0$ as its global minimizer. Furthermore, the lowest function value is given by $f(x_*) = 0$.

Suppose we want to start an iteration of the GS method with

$$x_0 = \left(0.5^l, 0.5^{2l}\right), \text{ for any fixed } l \in \mathbb{N}.$$

Moreover, we assume that the method has sampled in such a way that

$$f(x_{0,i}) = \phi_2(x_{0,i}),\ \forall i \in \{1,2,3\} \text{ (assuming } m = 3).$$

Consequently, the function ϕ_3 does not assume the maximum at the sampled points nor at x_0. Step 2 returns us $\tilde{g}_0 = (0,1)$. Assuming that $\nu_0 = \epsilon_0 = 10^{-1}$ and $\epsilon_{\text{opt}} = \nu_{\text{opt}} = 10^{-6}$, the method does not stop neither jumps from Step 3 to Step 6.

Now, notice that, for all $t > 0$, we have

$$\begin{aligned}
\phi_1(x_0 - t\tilde{g}_0) &= 0.5^{2l} + 0.5^l - t;\\
\phi_2(x_0 - t\tilde{g}_0) &= 0.5^{2l} - 2\cdot 0.5^l - t;\\
\phi_3(x_0 - t\tilde{g}_0) &= -2\cdot 0.5^{2l} + 0.5^l + 2t.
\end{aligned}$$

Hence, for $t = O(1)$, we must have that $f(x_0 - t\tilde{g}_0) = \phi_3(x_0 - t\tilde{g}_0)$, while for $t = \rho \cdot 0.5^{2l}$, with $\rho \in (0,1)$, we have $f(x_0 - t\tilde{g}_0) = \phi_1(x_0 - t\tilde{g}_0)$. Since $f(x_0) = -2\cdot 0.5^{2l} + 0.5^l$, we see clearly that for any $t_0 \in \{1, \gamma, \gamma^2, \ldots\}$, we must have

$$f(x_0) < \phi_3(x_0 - t_0\tilde{g}_0).$$

Consequently, in order to have a successful line search, we must be in a region of the domain where ϕ_3 does not assume the maximum, which is achieved by setting $t_0 = O(0.5^{2l})$.

Defining $x_1 = x_0 - t_0\tilde{g}_0$, one can compute the reduction efficiency of the function value, which yields

$$\frac{f(x_1) - f(x_*)}{f(x_0) - f(x_*)} = 1 + O(0.5^l).$$

Hence, one can see that it is not possible to establish a linear convergence rate no matter how close we start from x_* ($l \to \infty$).

Notice that the above example has the property that $U(x_*) = \{0\}$. We now present another example which has $U(x_*) \neq \{0\}$. So, let us consider a bidimensional function $f : \mathbb{R}^2 \to \mathbb{R}$, with

$$f(x) = \max\left\{\phi_1(x), \phi_2(x)\right\},$$

where, for $x = [\xi_1 \quad \xi_2]^T$, we have

$$\phi_1(x) = \frac{1}{2}\xi_1^2 - \xi_2 \text{ and } \phi_2(x) = \frac{1}{2}\xi_1^2 + \xi_2.$$

Clearly, f is a convex function with its optimal value being reached at $x_* = 0$.

Suppose we want to start an iteration of the GS method with

$$x_0 = \left[0.5^l \quad -0.5^{3l}\right]^T, \text{ for any fixed } l \in \mathbb{N}.$$

Moreover, assuming $m = 3$ in Algorithm 1 and taking any $\alpha > 0$, we suppose that the method has sampled in such a way that

$$f(x_{0,i}) = \phi_1(x_{0,i}) \text{ and } \|x_{0,i} - x_0\| \leqslant \alpha 0.5^{2l}, \text{ for all } i \in \{1, 2, 3\}.$$

Consequently, the function ϕ_2 does not assume the maximum at the sampled points nor at x_0. Therefore, by the way Step 2 is designed, Algorithm 1 returns $\tilde{g}_0 = [0.5^l + w_0 \quad -1]^T$, where $|w_0| \leqslant \alpha 0.5^{2l}$. Now, notice that, for all $t > 0$, we have

$$\phi_1(x_0 - t\tilde{g}_0) = \frac{1}{2}\left[0.5^l - t(0.5^l + w_0)\right]^2 + 0.5^{3l} - t;$$
$$\phi_2(x_0 - t\tilde{g}_0) = \frac{1}{2}\left[0.5^l - t(0.5^l + w_0)\right]^2 - 0.5^{3l} + t.$$

Hence, supposing that $\|\tilde{g}_0\| \geqslant \nu_0$, the GS method performs the line search procedure presented in Step 4. Then, noticing that

$$f(x_0) = \frac{1}{2}0.5^{2l} + 0.5^{3l},$$

we see that $t_0 = O(0.5^{3l})$, when one considers a sufficiently large l. Consequently, if $x_1 = x_0 - t_0\tilde{g}_0$, we have

$$\frac{f(x_1) - f(x_*)}{f(x_0) - f(x_*)} = 1 + O(0.5^l) \underset{l\to\infty}{\longrightarrow} 1.$$

These examples show that, when $k \notin \mathcal{K}$ (with $\mathcal{K}$ being the infinite index set of Assumption 3.1), the decrease of the function value may happen to be sublinear. It would be desirable to obtain a linear convergence result, since this convergence rate reinforces the idea that the GS method can be seen as a generalization of the Cauchy method. Therefore, it is reasonable to think that a local convergence result will rely on the condition that $k \in \mathcal{K}$. However, one might still wonder if H_ϕ is sufficient for our goal. Unfortunately, Assumption 3.1 is not enough for reaching our purposes, as the size of the sampling radius plays a key role as well (see Sections 3.2 and 3.3). Indeed, an additional condition must be taken into account: a restriction over the value of

$$\tau_k := \max_{1\leqslant i\leqslant m}\{\|x_{k,i} - x_k\|\}. \tag{3.2}$$

We state that assuming $\tau_k \leqslant T\|x_k - x_*\|^2$, for a sufficiently small $T > 0$, one can guarantee that, for iterations at which $k \in \mathcal{K}$, the function value will be reduced with a linear rate.

Before we proceed with the proofs, we need to make a remark. The local convergence theory developed here is applicable only to functions that have local minimizers x_* that satisfy $U(x_*) \neq \{0\}$. We justify this by looking at the function $f : \mathbb{R} \to \mathbb{R}$, stated as $f(x) = |x| = \max\{-x, x\}$. We have seen that for iterations that are not in $\mathcal{K}$, the GS method may present a very slow decrease of the function value. However, for the absolute value function, every time we have an iteration $k \in \mathcal{K}$, it yields that $\tilde{g}_k = 0$, and consequently, $x_{k+1} = x_k$. Therefore, the analysis for the case that $V(x_*) = \mathbb{R}^n$ cannot be based on H_ϕ and accordingly, from now on we assume that $U(x_*) \neq \{0\}$.

3.2 Convergence results

In this section, we establish a local convergence result for the nonnormalized version of the GS method. In other words, we find $R \in (0,1)$ such that, for infinitely many indices $k \in \mathbb{N}$, we have

$$f(x_{k+1}) - f(x_*) \leqslant R\,[f(x_k) - f(x_*)]\,.$$

To achieve our goal, we present an assumption that is commonly used in nonsmooth analysis [11, 36].

Assumption 3.2. *Let x_* be the local minimizer of f, previously exhibited at (3.1). The gradients $\{\nabla\phi_i(x_*)\}_{i\in\mathcal{I}(x_*)}$ compose an affinely independent set, that is,*

$$\sum_{i\in\mathcal{I}(x_*)} \alpha_i \nabla\phi_i(x_*) = 0 \quad \textit{and} \quad \sum_{i\in\mathcal{I}(x_*)} \alpha_i = 0 \iff \alpha_i = 0, \ \ \forall i \in \mathcal{I}(x_*).$$

In [22, Chapter III], we see that the aforementioned assumption guarantees, for any $j \in \mathcal{I}(x_*)$, that

$$\{\nabla\phi_i(x_*) - \nabla\phi_j(x_*)\}_{i\in\mathcal{I}(x_*)\setminus\{j\}}$$

is linearly independent. Consequently, $|\mathcal{I}(x_*)| \leqslant n+1$, and since we supposed $U(x_*) \neq \{0\}$, it implies that $|\mathcal{I}(x_*)| \leqslant n$.

The role played by Assumption 3.2 in our results is of great importance. It is worth noticing that the affine independence of the gradients $\nabla\phi_i(x_*)$ allows us to rule out redundant representations of the objective function near the local minimizer x_*. For instance, let us consider the bidimensional function used as the motivational example of the previous section. There is more than one way to write such a function, since one can also represent it as $f(\xi_1,\xi_2) = \max\{\xi_1^2+\xi_2, \xi_1^2-\xi_2, -\xi_1^2+\xi_2\}$. Obviously, $-\xi_1^2+\xi_2$ has no use in describing f. Moreover, there is no region near its optimal solution such that only $-\xi_1^2+\xi_2$ would assume the maximum, which cuts out the validity of H_ϕ. In fact, it is not difficult to see that redundant representations can be easily obtained for any objective function near a local minimizer. Therefore, when $U(x_*) \neq \{0\}$, Assumption 3.2 is imposing a good description of f (which does not need to be known) but not necessarily assuming a condition over the shape of f. Summing up, Assumption 3.2 brings some algebraic regularity to the optimization problem. One of the ways to see this regularity is by noticing that there is only one $\lambda^* \in \mathbb{R}^r$ that satisfies [22, Chapter III]

$$\lambda^* \geqslant 0, \quad \sum_{i=1}^{r} \lambda_i^* = 1 \quad \text{and} \quad \sum_{i=1}^{r} \lambda_i^* \nabla\phi_i(x_*) = 0. \tag{3.3}$$

Once we have elucidated some of the implications of Assumption 3.2, we are ready to present our first results. For this, we will consider that $|\mathcal{I}(x_*)| \geqslant 2$, since otherwise

the convergence would be to a point at which f is smooth, that is not the case of interest. Additionally, without any loss of generality, we assume that the neighborhood $\mathcal{W}$ presented in (3.1) is small enough such that the functions that comprise f in $\mathcal{W}$ are only the smooth functions that are active at x_* (which implies $r \leqslant n$). Moreover, for all $x \in \mathcal{W}$ and for any fixed $j \in \mathcal{I}(x_*)$,

$$\{\nabla\phi_i(x) - \nabla\phi_j(x)\}_{i\in\mathcal{I}(x_*)\setminus\{j\}}$$

is linearly independent.

We start our theoretical results with a technical lemma that will be directly linked to Assumption 3.3, presented subsequently.

Lemma 3.1. *Suppose Assumption 3.2 holds. Then, for any $d \in U(x_*)$, we must have*

$$d^T\left(\sum_{i=1}^{r}\lambda_i^*\nabla^2\phi_i(x_*)\right)d \geqslant 0.$$

Proof. Let us consider any vector $d \in U(x_*)$. Since $\phi_i \in C^2$, for all $i \in \{1,\dots,r\}$, and Assumption 3.2 holds, we can see by a cautious application of the Implicit Function Theorem [40, Appendix] that there exist a sufficiently small $\delta > 0$ and a twice differentiable function $\gamma : (-\delta,\delta) \to \mathbb{R}^n$ such that $\gamma(0) = x_*$, $\gamma'(0) = d$ and

$$t \in (-\delta,\delta) \Rightarrow \phi_i(\gamma(t)) - \phi_r(\gamma(t)) = 0, \quad \text{for all } i \in \{1,\dots,r-1\}.$$

Additionally, since x_* is a local minimizer of f, we must have that $t = 0$ is a local minimizer of the function $F(t) := \phi_r(\gamma(t))$. Consequently,

$$d^T\nabla^2\phi_r(x_*)d + \nabla\phi_r(x_*)^T\gamma''(0) = F''(0) \geqslant 0. \tag{3.4}$$

Now, defining $\psi_i(x) := \phi_i(x) - \phi_r(x)$, for $i \in \{1,\dots,r-1\}$, we must have

$$\psi_i(\gamma(t)) = \psi_i(x_*) + t\nabla\psi_i(x_*)^Td + \frac{t^2}{2}\left(d^T\nabla^2\psi_i(x_*)d + \nabla\psi_i(x_*)^T\gamma''(0)\right) + o(t^2).$$

Hence, since $\psi_i(\gamma(t)) = \psi_i(x_*) = 0$, for all $t \in (-\delta,\delta)$, and $\nabla\psi_i(x_*)^Td = 0$, for $i \in \{1,\dots,r-1\}$, we see, by taking the limit $t \to 0$, that

$$d^T\nabla^2\psi_i(x_*)d + \nabla\psi_i(x_*)^T\gamma''(0) = 0, \ \ \forall i \in \{1,\dots,r-1\},$$

which yields

$$d^T\sum_{i=1}^{r-1}\lambda_i^*\left[\nabla^2\phi_i(x_*) - \nabla^2\phi_r(x_*)\right]d + \sum_{i=1}^{r-1}\lambda_i^*\left[\nabla\phi_i(x_*) - \nabla\phi_r(x_*)\right]^T\gamma''(0) = 0.$$

Finally, adding the last equation to (3.4) and recalling that $e^T\lambda^* = 1$, we have

$$d^T\left(\sum_{i=1}^{r}\lambda_i^*\nabla^2\phi_i(x_*)\right)d + \sum_{i=1}^{r}\lambda_i^*\nabla\phi_i(x_*)^T\gamma''(0) \geqslant 0,$$

which implies the desired result (because $\sum_{i=1}^{r}\lambda_i^*\nabla\phi_i(x_*) = 0$). □

The result presented above is a strong statement. It highlights that the positive semidefinite U-Hessian matrix

$$\sum_{i=1}^{r} \lambda_i^* \nabla^2 \phi_i(x_*)$$

will play the role of a generalized Hessian of the function f in the U-space. Now, following closely the analysis made in [37, Section 5.1] for the case where f is a convex function, we motivate our next assumption.

Assumption 3.3. *The local minimizer x_* of f is a strong minimizer*[1]*, i.e., $0 \in ri\,\overline{\partial} f(x_*)$ and there is $\mu > 0$ such that*

$$d^T \left(\sum_{i=1}^{r} \lambda_i^* \nabla^2 \phi_i(x_*) \right) d \geqslant \mu \|d\|^2, \quad \text{for all } d \in U(x_*). \tag{3.5}$$

Remark 3.1. *Under the condition $0 \in ri\,\overline{\partial} f(x_*)$ and Assumption 3.2, one can see that the unique vector that satisfies (3.3) must have strictly positive entries, i.e., $\lambda^* > 0$ (see [22, Remark III.2.1.4]). Consequently, if H_ϕ does not hold, then, in the case that $x_k \to x_*$, we must have $\|\tilde{g}_k\|$ bounded away from zero for any sufficiently large k. However, [27, Theorem 3.3] guarantees, with probability one, that there is an infinite index set $\mathcal{K} \subset \mathbb{N}$ such that $\|\tilde{g}_k\| \underset{k \in \mathcal{K}}{\to} 0$ (since $\epsilon_k, \nu_k \to 0$). Hence, this implies that under the Assumptions 3.2 and 3.3 and supposing $x_k \to x_*$, the probability that Assumption 3.1 does not hold is zero. Therefore, although Assumptions 3.2 and 3.3 are sufficient to have Assumption 3.1 (with probability one), for clarity of our results, we present Assumption 3.1 as a hypothesis, which exempts us from obtaining statements involving probability.*

The next lemma ensures a sufficient growth of f at x_* for directions close enough to the subspace $U(x_*)$.

Lemma 3.2. *Suppose that Assumptions 3.2 and 3.3 hold. Moreover, let us assume a sequence $\{s_k\} \subset \mathbb{R}^n$ such that $s_k \to 0$ and* [2]

$$s_k = \mathrm{Proj}_{U(x_*)}(s_k) + o(\|s_k\|). \tag{3.6}$$

Then, for all sufficiently large $k \in \mathbb{N}$, the following must be valid

i) $s_k^T \left(\sum_{i=1}^{r} \lambda_i^* \nabla^2 \phi_i(x_*) \right) s_k \geqslant \frac{2}{3} \mu \|s_k\|^2$;

ii) $f(x_*) + \frac{\mu}{4} \|s_k\|^2 \leqslant f(x_* + s_k)$,

where μ is such that (3.5) holds.

[1] The property that $0 \in \mathrm{ri}\,\overline{\partial} f(x_*)$ is also called *nondegeneracy* [11].

Proof. First, let us prove statement i). Since

$$\left\|\mathrm{Proj}_{U(x_*)}(s_k)\right\| \leqslant \|s_k\|,$$

notice that

$$\begin{aligned} s_k^T\left(\sum_{i=1}^r \lambda_i^*\nabla^2\phi_i(x_*)\right)s_k \geqslant\ & \mathrm{Proj}_{U(x_*)}(s_k)^T\left(\sum_{i=1}^r \lambda_i^*\nabla^2\phi_i(x_*)\right)\mathrm{Proj}_{U(x_*)}(s_k) \\ & - 2\left\|\mathrm{Proj}_{U(x_*)}(s_k)\right\| o(\|s_k\|) + o(\|s_k\|^2) \\ \geqslant\ & \mu\left\|\mathrm{Proj}_{U(x_*)}(s_k)\right\|^2 + o(\|s_k\|^2) \\ =\ & \mu\|s_k\|^2 + o(\|s_k\|^2) \\ & \text{(by relation (3.6)).} \end{aligned}$$

Hence, for a sufficiently large $k \in \mathbb{N}$, the first result is obtained.

Now, let us prove the second statement. Notice that, for a sufficiently large $k \in \mathbb{N}$, we must have

$$\begin{aligned} f(x_* + s_k) =\ & \max_{1\leqslant i\leqslant r}\{\phi_i(x_* + s_k)\} \\ \geqslant\ & \sum_{i=1}^r \lambda_i^*\phi_i(x_* + s_k) \\ =\ & \sum_{i=1}^r \lambda_i^*\left[\phi_i(x_*) + \nabla\phi_i(x_*)^T s_k + \frac{1}{2}s_k^T\nabla^2\phi_i(x_*)s_k\right] + o(\|s_k\|^2) \\ =\ & f(x_*) + \frac{1}{2}s_k^T\left(\sum_{i=1}^r \lambda_i^*\nabla^2\phi_i(x_*)\right)s_k + o(\|s_k\|^2). \end{aligned}$$

Consequently,

$$\frac{f(x_* + s_k) - f(x_*)}{\|s_k\|^2} \geqslant \frac{1}{2}\frac{s_k^T}{\|s_k\|}\left(\sum_{i=1}^r \lambda_i^*\nabla^2\phi_i(x_*)\right)\frac{s_k}{\|s_k\|} + \frac{o(\|s_k\|^2)}{\|s_k\|^2}.$$

Recalling the inequality that appears in i), we obtain

$$\frac{f(x_* + s_k) - f(x_*)}{\|s_k\|^2} \geqslant \frac{1}{3}\mu + \frac{o(\|s_k\|^2)}{\|s_k\|^2}.$$

Hence, for all $k \in \mathbb{N}$ sufficiently large, we must have

$$\frac{f(x_* + s_k) - f(x_*)}{\|s_k\|^2} \geqslant \frac{1}{4}\mu,$$

which is the desired result. □

[2] Here we have an abuse of notation. The relation (3.6) stands for $s_k = \mathrm{Proj}_{U(x_*)}(s_k) + v_k$, where $\|v_k\| = o(\|s_k\|)$. This notation will appear in other parts of the text.

We have seen in Remark 3.1 that, when $x_k \to x_*$, Assumption 3.1 is a necessary condition to

$$\liminf_{k\to\infty} \|\tilde{g}_k\| = 0 \tag{3.7}$$

be valid. The next lemma guarantees sufficient conditions for (3.7) to hold.

Lemma 3.3. *Suppose that Assumptions 3.1 and 3.2 are verified. Then,*

$$\tilde{g}_k \underset{k\in\mathcal{K}}{\to} 0 \quad \textit{and} \quad \hat{\lambda}^k \underset{k\in\mathcal{K}}{\to} \lambda^*,$$

where $\hat{\lambda}^k \in \mathbb{R}^r$ *and, for* $i \in \{1,\dots,r\}$,

$$\hat{\lambda}^k_i := \sum_{j\in\mathcal{J}_{k,i}} \lambda^k_j, \quad \textit{with } \mathcal{J}_{k,i} := \{s \in \{0,\dots,m\} \mid f(x_{k,s}) = \phi_i(x_{k,s})\}. \tag{3.8}$$

Proof. Assuming $k \in \mathcal{K}$, it yields $\mathcal{B}(x_k,\epsilon_k) \subset \mathcal{W}$ and that, for each $i \in \{1,\dots,r\}$, $\mathcal{J}_{k,i}$ is not empty. So, recalling the definition of τ_k in (3.2) and that λ^k solves the quadratic minimization problem of Step 2, we get

$$\begin{aligned}
\|\tilde{g}_k\| &= \left\| \sum_{i=1}^{r} \sum_{j\in\mathcal{J}_{k,i}} \lambda^k_j \nabla\phi_i(x_{k,j}) \right\| \\
&\leqslant \left\| \sum_{i=1}^{r} \sum_{j\in\mathcal{J}_{k,i}} \frac{\lambda^*_i}{|\mathcal{J}_{k,i}|} \nabla\phi_i(x_{k,j}) \right\| \\
&= \left\| \sum_{i=1}^{r} \sum_{j\in\mathcal{J}_{k,i}} \frac{\lambda^*_i}{|\mathcal{J}_{k,i}|} \nabla\phi_i(x_k) \right\| + O(\tau_k) \\
&= \left\| \sum_{i=1}^{r} \lambda^*_i \nabla\phi_i(x_k) \right\| + O(\tau_k).
\end{aligned} \tag{3.9}$$

Hence, since $x_k \to x_*$, $\epsilon_k \to 0$ and $\tau_k \in (0,\epsilon_k)$, we obtain

$$\tilde{g}_k \underset{k\in\mathcal{K}}{\to} 0.$$

Moreover, it implies

$$\tilde{g}_k = \sum_{i=1}^{r} \sum_{j\in\mathcal{J}_{k,i}} \lambda^k_j \nabla\phi_i(x_k) + O(\tau_k) = \sum_{i=1}^{r} \hat{\lambda}^k_i \nabla\phi_i(x_k) + O(\tau_k) \underset{k\in\mathcal{K}}{\to} 0.$$

Now, since Assumption 3.2 holds and $\lambda^* \in \mathbb{R}^r$ is the unique vector satisfying (3.3), we must have

$$\hat{\lambda}^k \underset{k\in\mathcal{K}}{\to} \lambda^*,$$

which ends the proof. □

The next technical lemma establishes sufficient conditions ensuring that the vector $x_k - x_*$ will be close enough to the subspace $U(x_*)$.

Lemma 3.4. *Suppose that Assumption 3.1 and 3.2 hold. Then, there must exist $k' \in \mathcal{K}$, such that for all $k \in \mathcal{K}$ larger than k' and having $\tau_k \leqslant \alpha\|x_k - x_*\|^2$, for any fixed $\alpha > 0$, the following must happen*

i) For all $i \in \{1, \ldots, r-1\}$, we have

$$|\phi_i(x_k) - \phi_r(x_k)| \leqslant 2\alpha L_{max}\|x_k - x_*\|^2,$$

where L_{max} is an upper bound for the Lipschitz constants of the functions ϕ_i around x_*;

ii) $x_k - x_* = \text{Proj}_{U(x_*)}(x_k - x_*) + o(\|x_k - x_*\|)$.

Proof. First, since Assumption 3.1 holds, there are points $y_1, \ldots, y_r \in \overline{\mathcal{B}}(x_k, \tau_k)$, for all $k \in \mathcal{K}$, such that

$$\phi_r(y_r) > \phi_i(y_r) \quad \text{and} \quad \phi_r(y_i) < \phi_i(y_i), \quad i \in \{1, \ldots, r-1\}.$$

Therefore, defining $\psi_i := \phi_i - \phi_r$, we have, by the Intermediate Value Theorem, that there exists $z_i \in \overline{\mathcal{B}}(x_k, \tau_k)$ such that $\psi_i(z_i) = 0$, for all $i \in \{1, \ldots, r-1\}$. Consequently, considering $k \in \mathcal{K}$ and $L_{\max}$ as a valid upper bound for the Lipschitz constants of the functions ϕ_i in $\mathcal{W}$, the following holds

$$\begin{aligned}\phi_i(z_i) = \phi_r(z_i) \Rightarrow |\phi_i(x_k) - \phi_r(x_k)| &= |\phi_i(x_k) - \phi_i(z_i) + \phi_r(z_i) - \phi_r(x_k)| \\ &\Rightarrow |\phi_i(x_k) - \phi_r(x_k)| \leqslant 2L_{\max}\tau_k.\end{aligned}$$

Since $\tau_k \leqslant \alpha\|x_k - x_*\|^2$, the first result is obtained.

Now, let us consider the Taylor's expansion of the functions ϕ_i, with $i \in \{1, \ldots, r\}$. Then,

$$\phi_i(x_k) = \phi_i(x_*) + \nabla\phi_i(x_*)^T(x_k - x_*) + O(\|x_k - x_*\|^2).$$

So, for $i \in \{1, \ldots, r-1\}$,

$$\phi_i(x_k) - \phi_r(x_k) = \left[\nabla\phi_i(x_*) - \nabla\phi_r(x_*)\right]^T(x_k - x_*) + O(\|x_k - x_*\|^2),$$

which yields, by item i), that

$$\left[\nabla\phi_i(x_*) - \nabla\phi_r(x_*)\right]^T(x_k - x_*) = O(\|x_k - x_*\|^2).$$

Therefore, because of the definition of the subspace $U(x_*)$, we must have

$$x_k - x_* = \text{Proj}_{U(x_*)}(x_k - x_*) + o(\|x_k - x_*\|),$$

as desired. □

The following statement says that, under some hypotheses, the difference $f(x_k) - f(x_*)$ can be bounded above by a value proportional to $\|\tilde{g}_k\|^2$. This and the other subsequent results pursuit, for the nonsmooth case, equivalent statements of the well established local convergence result of the steepest descent method [2, Chapter 2]. For this goal, the hypothesis about the value τ_k in Lemma 3.5 turns to be essential.

Lemma 3.5. *Suppose that Assumptions 3.1, 3.2 and 3.3 hold. Then, there must exist $k' \in \mathcal{K}$, such that*

$$k \in \mathcal{K},\ \textit{with}\ k \geqslant k',\ \textit{and}\ \tau_k \leqslant \frac{\mu}{8L_{max}}\|x_k - x_*\|^2 \Rightarrow \frac{\mu}{4}\left[f(x_k) - f(x_*)\right] \leqslant \|\tilde{g}_k\|^2.$$

Proof. Let us consider $k \in \mathcal{K}$ and that

$$\tau_k \leqslant \frac{\mu}{8L_{\max}}\|x_k - x_*\|^2.$$

Now, using the definition of $\hat{\lambda}^k$ in (3.8), one can notice that

$$\begin{aligned}
f(x_*) &= \max_{1\leqslant i\leqslant r}\{\phi_i(x_*)\} \\
&= \max_{1\leqslant i\leqslant r}\left\{\phi_i(x_k) + \nabla\phi_i(x_k)^T(x_* - x_k) + \frac{1}{2}(x_* - x_k)^T\nabla^2\phi_i(x_k)(x_* - x_k)\right\} \\
&\quad + o(\|x_k - x_*\|^2) \\
&\geqslant \sum_{i=1}^{r}\hat{\lambda}_i^k\left[\phi_i(x_k) + \nabla\phi_i(x_k)^T(x_* - x_k) + \frac{1}{2}(x_* - x_k)^T\nabla^2\phi_i(x_k)(x_* - x_k)\right] \\
&\quad + o(\|x_k - x_*\|^2).
\end{aligned}$$

Assuming, without loss of generality, that

$$\max_{1\leqslant i\leqslant r}\{\phi_i(x_k)\} = \phi_r(x_k)$$

and recalling i) of Lemma 3.4, we have

$$\begin{aligned}
\sum_{i=1}^{r}\hat{\lambda}_i^k\phi_i(x_k) &\geqslant \max_{1\leqslant i\leqslant r}\{\phi_i(x_k)\} - \frac{\mu}{8L_{\max}}2L_{\max}\|x_k - x_*\|^2 \\
&= f(x_k) - \frac{\mu}{4}\|x_k - x_*\|^2.
\end{aligned}$$

Additionally, since the derivatives of ϕ_i are all Lipschitz continuous, we must have

$$\begin{aligned}
\sum_{i=1}^{r}\hat{\lambda}_i^k\nabla\phi_i(x_k)^T(x_* - x_k) &= \sum_{i=1}^{r}\sum_{j\in\mathcal{J}_{k,i}}\lambda_j^k\nabla\phi_i(x_k)^T(x_* - x_k) \\
&= \sum_{i=1}^{r}\sum_{j\in\mathcal{J}_{k,i}}\lambda_j^k\nabla\phi_i(x_{k,j})^T(x_* - x_k) + o(\|x_k - x_*\|^2) \\
&= \tilde{g}_k^T(x_* - x_k) + o(\|x_k - x_*\|^2).
\end{aligned}$$

By the fact that the functions ϕ_i are of class C^2 and by Lemmas 3.2, 3.3 and 3.4, we see, for a sufficiently large $k \in \mathcal{K}$, that

$$\begin{aligned}(x_* - x_k)^T \sum_{i=1}^{r} \hat{\lambda}_i^k \nabla^2 \phi_i(x_k)(x_* - x_k) &= (x_* - x_k)^T \sum_{i=1}^{r} \lambda_i^* \nabla^2 \phi_i(x_*)(x_* - x_k) \\ &\quad + o(\|x_k - x_*\|^2) \\ &\geqslant \frac{2}{3}\mu\|x_k - x_*\|^2 + o(\|x_k - x_*\|^2).\end{aligned}$$

Therefore, the following must hold

$$\begin{aligned}f(x_*) &\geqslant f(x_k) + \tilde{g}_k^T(x_* - x_k) + \frac{2}{3}\mu\|x_k - x_*\|^2 \\ &\quad - \frac{1}{4}\mu\|x_k - x_*\|^2 + o(\|x_k - x_*\|^2) \\ &= f(x_k) + \tilde{g}_k^T(x_* - x_k) + \frac{5}{12}\mu\|x_k - x_*\|^2 + o(\|x_k - x_*\|^2).\end{aligned}$$

Consequently, for $k \in \mathcal{K}$ sufficiently large, it yields

$$f(x_k) - f(x_*) \leqslant \tilde{g}_k^T(x_k - x_*) \leqslant \|\tilde{g}_k\|\|x_k - x_*\|. \tag{3.10}$$

By Lemma 3.4, we see that, for a sufficiently large $k \in \mathcal{K}$, the hypotheses of Lemma 3.2 will hold for $s_k = x_k - x_*$. So, *ii)* of Lemma 3.2 implies

$$\|x_k - x_*\| \leqslant 2\sqrt{\frac{f(x_k) - f(x_*)}{\mu}}.$$

Finally, from the last inequality and (3.10), we obtain the desired result. □

The result below guarantees a sufficient decrease of the function value. This lemma will be of great importance in our main theorem.

Lemma 3.6. *Suppose that Assumptions 3.1, 3.2 and 3.3 hold. Then, there must exist $k' \in \mathcal{K}$, such that for all $k \in \mathcal{K}$ larger than k' with*

$$\tau_k \leqslant \frac{\mu}{8L_{max}}\|x_k - x_*\|^2, \tag{3.11}$$

the following must happen

$$\gamma\frac{1-\beta}{M} \leqslant t \leqslant \frac{1-\beta}{M} \Rightarrow f(x_k - t\tilde{g}_k) < f(x_k) - \beta t\|\tilde{g}_k\|^2,$$

where M is a positive real number such that

$$\max_{1\leqslant i\leqslant r}\left\{\|\nabla^2\phi_i(x)\|\right\} \leqslant M, \quad \textit{for all } x \in \mathcal{W}. \tag{3.12}$$

Proof. Let us consider an index $k \in \mathcal{K}$ sufficiently large and that τ_k satisfies the upper bound of (3.11). Then, considering a fixed $t \in (0, 1]$, we have

$$\begin{aligned} f(x_k - t\tilde{g}_k) = & \max_{1\leqslant i\leqslant r}\left\{\phi_i(x_k) - t\nabla\phi_i(x_k)^T\tilde{g}_k + \frac{t^2}{2}\tilde{g}_k^T\nabla^2\phi_i(x_k)\tilde{g}_k\right\} \\ & + o(\|\tilde{g}_k\|^2) \\ \leqslant & f(x_k) + \max_{1\leqslant i\leqslant r}\left\{-t\nabla\phi_i(x_k)^T\tilde{g}_k\right\} \\ & + \frac{t^2}{2}\max_{1\leqslant i\leqslant r}\left\{\tilde{g}_k^T\nabla^2\phi_i(x_k)\tilde{g}_k\right\} + o(\|\tilde{g}_k\|^2). \end{aligned}$$

Additionally, choosing $k \in \mathcal{K}$ large enough and considering $s_k = x_k - x_*$ in $ii)$ of Lemma 3.2 and the result of Lemma 3.5, we see that $\tau_k = O(\|\tilde{g}_k\|^2)$. Consequently,

$$\max_{1\leqslant i\leqslant r}\left\{-t\nabla\phi_i(x_k)^T\tilde{g}_k\right\} = \max_{0\leqslant i\leqslant m}\left\{-t\nabla f(x_{k,i})^T\tilde{g}_k\right\} + o(\|\tilde{g}_k\|^2).$$

Moreover, from convex analysis [22, Chapter 3], we know that

$$\tilde{g}_k \ \text{ solves } \min_{g\in\mathrm{co}\{\nabla f(x_{k,i})\}_{i=0}^m}\|g\| \Leftrightarrow \langle g - \tilde{g}_k, -\tilde{g}_k\rangle \leqslant 0,\ \forall g \in \mathrm{co}\{\nabla f(x_{k,i})\}_{i=0}^m,$$

which yields

$$\max_{0\leqslant i\leqslant m}\left\{-t\nabla f(x_{k,i})^T\tilde{g}_k\right\} \leqslant -t\|\tilde{g}_k\|^2.$$

Hence, it implies

$$f(x_k - t\tilde{g}_k) \leqslant f(x_k) - t\|\tilde{g}_k\|^2 + \frac{t^2}{2}\max_{1\leqslant i\leqslant r}\left\{\tilde{g}_k^T\nabla^2\phi_i(x_k)\tilde{g}_k\right\} + o(\|\tilde{g}_k\|^2).$$

Moreover, since $x_k \to x_*$, we must have

$$\max_{1\leqslant i\leqslant r}\left\{\tilde{g}_k^T\nabla^2\phi_i(x_k)\tilde{g}_k\right\} \leqslant M\|\tilde{g}_k\|^2, \ \text{ for all } x_k \text{ close enough to } x_*.$$

Therefore, since $\tilde{g}_k$ tends to the null vector for indices in $\mathcal{K}$ (by Lemma 3.3), there must exist a sufficiently large $k' \in \mathcal{K}$ such that for all $k \in \mathcal{K}$ larger than k' and

$$\gamma\frac{1-\beta}{M} \leqslant t \leqslant \frac{1-\beta}{M},$$

we have

$$\begin{aligned} f(x_k - t\tilde{g}_k) &< f(x_k) - t\|\tilde{g}_k\|^2 + t^2M\|\tilde{g}_k\|^2 \\ &= f(x_k) - t\|\tilde{g}_k\|^2(1 - Mt) \\ &\leqslant f(x_k) - \beta t\|\tilde{g}_k\|^2, \end{aligned}$$

which completes the proof. □

Finally, we are able to prove our main result. It establishes, under special conditions, that the GS method, in fact, has a linear convergence rate for some iterations.

Theorem 3.1. *Suppose that Assumptions 3.1, 3.2 and 3.3 hold. Then, there must exist $k' \in \mathcal{K}$, such that*

$$k \in \mathcal{K} \text{ with } k \geqslant k',\ \tau_k \leqslant \frac{\mu}{8L_{max}}\|x_k - x_*\|^2 \text{ and } x_{k+1} = x_k - t_k\tilde{g}_k, \tag{3.13}$$

implies

$$f(x_{k+1}) - f(x_*) \leqslant \left(1 - \mu\gamma\frac{\beta(1-\beta)}{4M}\right)[f(x_k) - f(x_*)].$$

Proof. First, let us suppose that we have $k' \in \mathcal{K}$ large enough so that Lemmas 3.5 and 3.6 hold. Then, assuming $k \geqslant k'$ and (3.13), one can notice that since t_k is obtained using Step 4 of Algorithm 1, we must have, by Lemma 3.6, that

$$t_k \geqslant \gamma\frac{1-\beta}{M}.$$

Therefore,

$$f(x_{k+1}) < f(x_k) - \beta t_k\|\tilde{g}_k\|^2 \leqslant f(x_k) - \gamma\frac{\beta(1-\beta)}{M}\|\tilde{g}_k\|^2.$$

Consequently, by Lemma 3.5, we obtain

$$f(x_{k+1}) - f(x_k) \leqslant -\gamma\frac{\beta(1-\beta)}{M}\frac{\mu}{4}[f(x_k) - f(x_*)],$$

which yields

$$f(x_{k+1}) - f(x_*) \leqslant \left(1 - \mu\gamma\frac{\beta(1-\beta)}{4M}\right)[f(x_k) - f(x_*)],$$

as desired. □

3.3 Practical implications

By the results from the last section, we see that an essential hypothesis to obtain those statements is

$$\tau_k \leqslant \frac{\mu}{8L_{\max}}\|x_k - x_*\|^2. \tag{3.14}$$

By Step 2, the probability of such a condition to hold in any fixed $k \in \mathbb{N}$ is directly linked to the value of ϵ_k. If ϵ_k is significantly larger than the upper bound required for τ_k, then the probability of (3.14) to happen is low. On the other hand, if ϵ_k is small enough, such a condition has a high probability to hold.

At least theoretically, we have a strong reason to ask for

$$\epsilon_k \approx \frac{\mu}{8L_{\max}}\|x_k - x_*\|^2.$$

Unfortunately, the knowledge of μ and $L_{\max}$ is not a reality for most of the problems. Moreover, x_* is the ultimate goal of GS, which implies that $\|x_k - x_*\|$ cannot be directly

computed. Therefore, it seems difficult to guarantee this approximation. Fortunately, such a requirement is not impossible to be satisfied in practice.

Indeed, let us consider the infinite index set $\mathcal{K} \subset \mathbb{N}$ presented in Assumption 3.1. Then, by (3.9), we see, for $k \in \mathcal{K}$, that

$$\|\tilde{g}_k\| \leqslant \left\| \sum_{i=1}^{r} \lambda_i^* \nabla \phi_i(x_k) \right\| + O(\tau_k).$$

For now, let us assume that, for all $k \in \mathbb{N}$, we know how to guarantee $\tau_k = O(\|\tilde{g}_k\|^{2+\rho})$, for some fixed $\rho > 0$. Then,

$$\begin{aligned} \|\tilde{g}_k\| \left[1 + O(\|\tilde{g}_k\|^{1+\rho})\right] \leqslant & \left\| \sum_{i=1}^{r} \lambda_i^* \nabla \phi_i(x_k) \right\| \\ = & \left\| \sum_{i=1}^{r} \lambda_i^* \left[\nabla \phi_i(x_*) + \nabla^2 \phi_i(x_*)(x_k - x_*)\right] \right\| \\ & + o(\|x_k - x_*\|). \end{aligned}$$

Consequently, since $\sum_{i=1}^{r} \lambda_i^* \nabla \phi_i(x_*) = 0$, we obtain $\|\tilde{g}_k\| = O(\|x_k - x_*\|)$. Recalling that $\tau_k = O(\|\tilde{g}_k\|^{2+\rho})$, it yields

$$\tau_k = O(\|x_k - x_*\|^{2+\rho}), \quad k \in \mathcal{K}.$$

Therefore, for any sufficiently large $k \in \mathcal{K}$, we have (3.14), which is our desired hypothesis.

The only gap that we have left out is how to ensure $\tau_k = O(\|\tilde{g}_k\|^{2+\rho})$. To this aim, we just need to set the following adjustment in Step 0:

$$\theta_\epsilon = (\theta_\nu)^{2+\rho}, \text{ for any desired value of } \rho > 0. \tag{3.15}$$

Indeed, defining l_k as the number of times the algorithm has reduced the sampling radius until the iteration k, and assuming that $x_{k+1} = x_k - t_k \tilde{g}_k$ (i.e., $x_k - t_k \tilde{g}_k \in \mathcal{D}$), we have

$$\tau_k \leqslant \epsilon_k = (\theta_\epsilon)^{l_k} \epsilon_0 = \left[(\theta_\nu)^{l_k}\right]^{2+\rho} \epsilon_0 = \left(\frac{\nu_k}{\nu_0}\right)^{2+\rho} \epsilon_0.$$

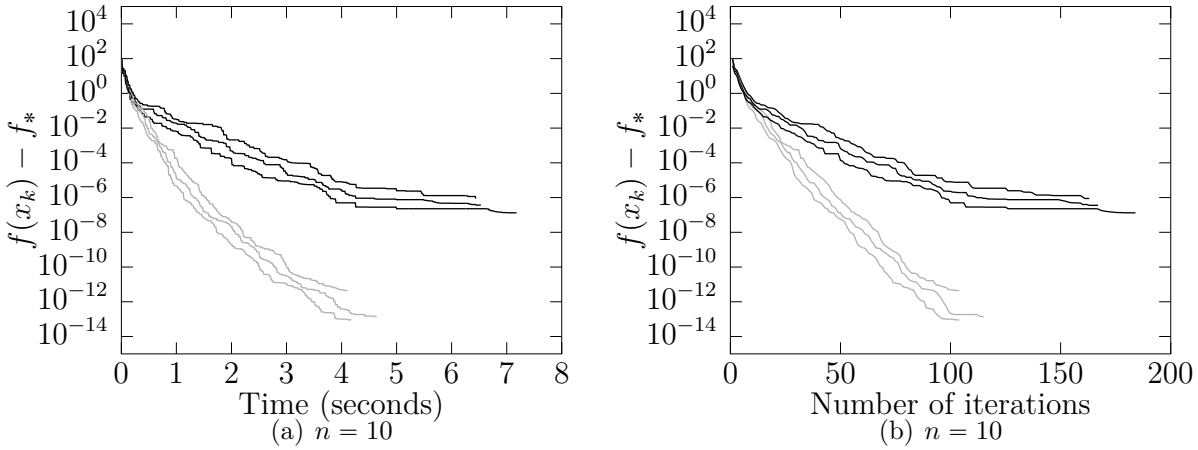
Hence, since $x_{k+1} = x_k - t_k \tilde{g}_k$, it means that a line search procedure was performed and, therefore, we must have $\|\tilde{g}_k\| \geqslant \nu_k$, which guarantees $\tau_k = O(\|\tilde{g}_k\|^{2+\rho})$.

As a result, (3.15) gives a practical implication for the GS method. In fact, what one really needs to ask is that $\epsilon_k = O(\nu_k^{2+\rho})$, for all sufficiently large k. The equality (3.15) is just a way to ensure this relation between ϵ_k and ν_k. To the best of our knowledge, there is no previous study that uses theoretical arguments to help a potential user to set the parameter values of θ_ν and θ_ϵ.

Finally, we present two illustrative examples in order to stress the importance of relation (3.14) and one to show the behavior of our approach when some assumptions do not hold. We compare the number of iterations and time (in seconds) versus the distance of the current function value to the minimum function value f_* reached along twenty runs. For each example, we exhibit the median with the first and third quartiles of those runs. The curves in black stand for the GS method with the parameter values suggested by the original authors [6], whereas the grey curves represent the same GS method but now using:

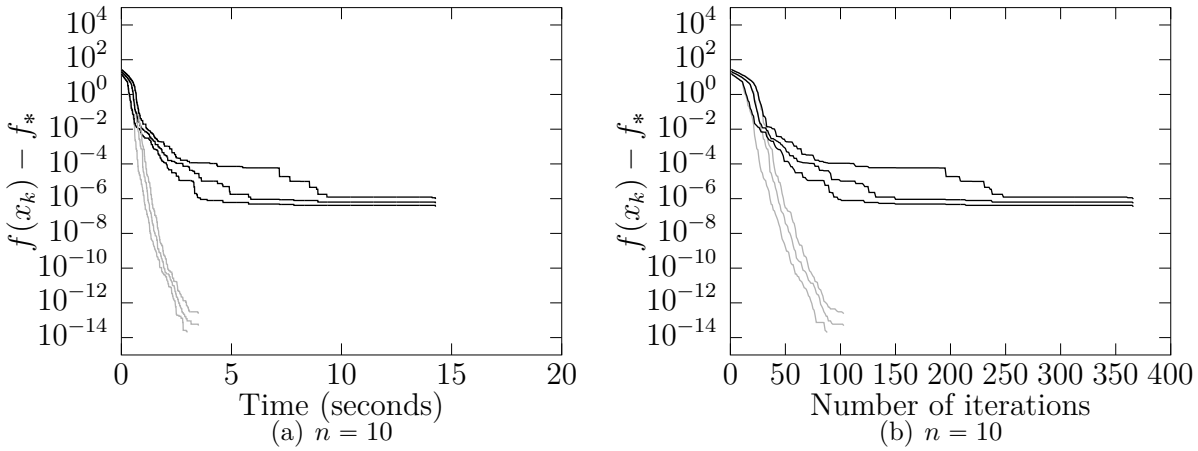
$$\nu_k = 10^{-(l_k+1)};\ \epsilon_0 = \nu_0,\ \epsilon_k = \nu_k^{1.5}, \text{ if } l_k = 1, \text{ and } \epsilon_k = \nu_k^{2.25}, \text{ if } l_k \geqslant 2.$$

All the results were obtained using `Matlab` and its function `quadprog` for addressing the GS subproblem of Step 2.



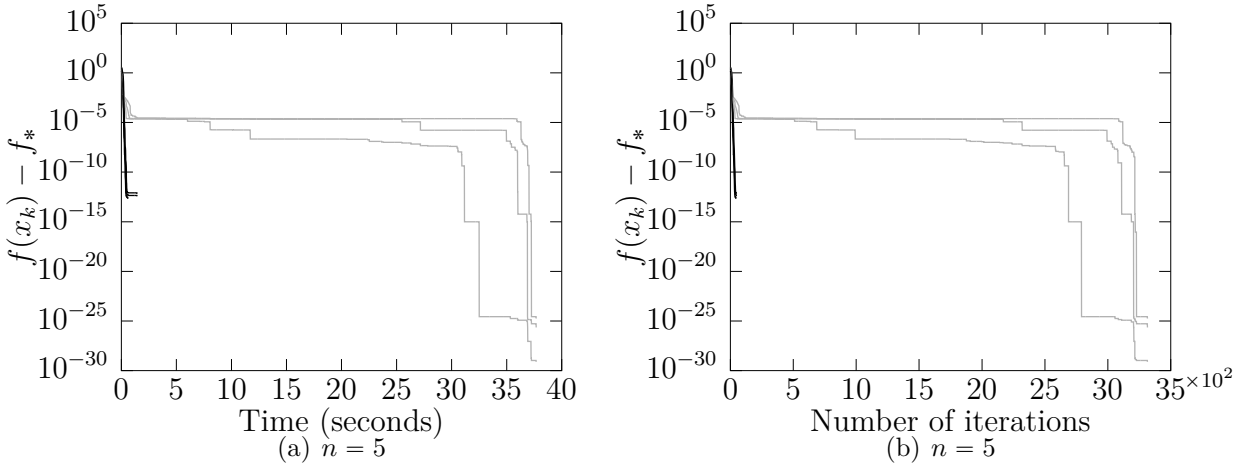
(a) $n = 10$ (b) $n = 10$

Figure 13 – Results for the nonsmooth convex function `Chained CB3 II` [47]. It satisfies $U(x_*) \neq \{0\}$.



(a) $n = 10$ (b) $n = 10$

Figure 14 – Results for the nonsmooth nonconvex function `Chained Crescent I` [47]. It satisfies $U(x_*) \neq \{0\}$.



(a) $n = 5$ (b) $n = 5$

Figure 15 – Results for the nonsmooth convex function `MAXQ` [47]. It does not satisfy Assumption 3.2.

In Figure 1, we have minimized the convex function `Chained CB3 II` [47]. This objective function presents to have $\dim V(x_*) = 2$ and, consequently, $\dim U(x_*) = 8$. Because the parameter values of the GS method, whose results are depicted in the black curves, do not ensure relation (3.14) to hold, the sampled points start to be too far from the current iterate, preventing the GS method from sustaining a good decreasing rate of the function value. On the other hand, when the parameter values are chosen in a manner that τ_k becomes small enough, the decreasing rate is preserved in most of the iterations.

The same behavior can be seen for the function that is minimized in Figure 2. Although, this objective function is nonconvex, we can see by Lemma 3.1 that restrained to the subspace where f behaves smoothly (for this case, we have $\dim U(x_*) = 9$), the generalized Hessian is positive semidefinite. Therefore, since Lemma 3.4 guarantees that x_k approaches the optimal point x_* by tracking the subspace $U(x_*)$ (for some specific iterations), the GS method starts to follow closely the behavior of the steepest descent method. But again, the value τ_k plays a key role on the rate of convergence.

Finally, Figures 3 shows a bad behavior of the GS method when one considers our approach. This can be explained by the lack of validity of Assumptions 3.2 and 3.3. For this case, we have

$$f(x) = \max_{1 \leqslant i \leqslant n} \{x_i^2\}.$$

Therefore, one can see that the gradients of the smooth functions that comprise f are not affinely independent, which does not ensure the uniqueness of the vector λ^*. Moreover, $0 \notin \operatorname{ri} \overline{\partial} f(x_*)$, and therefore, x_* is not a strong minimizer for the objective function. Consequently, the theoretical results cannot support a linear convergence rate of the functions values, a fact that can also be seen in practice.

3.4 Discussion

In this study, we have shown a linear local convergence result for the function value sequence generated by the nonnormalized version of the GS method. Our analysis does not provide any kind of local convergence result for functions with $V(x_*) = \mathbb{R}^n$. Moreover, as it is reasonable to expect, for nonsmooth functions satisfying $U(x_*) \neq \{0\}$, a good decrease of the function value is strongly dependent on a good set of sampled points. This set needs to cover all functions ϕ_i near x_*. More than that, a restriction over the size of τ_k is also a crucial hypothesis.

Although the assumption over τ_k seems impracticable to be verified, we have shown that such a requirement can be satisfied by tunning properly the values of the parameters θ_ν and θ_ϵ. We believe this is an important implication, since as far we are concerned, there is no previous theoretical argument that corroborates any particular choice of such parameters.

In conclusion, this study reinforces what was already a belief in the nonsmooth field, by giving a theoretical proof and establishing in which circumstances one can expect linear local convergence of the GS method.

4 Gradient and Function Sampling Method

Although the GS method has shown to be robust, presenting good numerical results even for challenging problems, its computational cost can be an obstacle, specially if we take into account the expectation over the rate of convergence. Therefore, this leads to a natural question: would it be possible to have a GS algorithm that can be understood as a generalization of Newton's (or quasi-Newton) method for nonsmooth functions, meaning that it would locally converge faster than linearly?

This study has the intent to start answering this question. As we shall see, the answer is, at least, partially affirmative. In fact, there are recent studies that have introduced GS-like algorithms with quasi-Newton techniques [9, 10], however there are no proofs nor numerical results that corroborate a rapid local convergence. Therefore, our affirmative answer is directly linked to the property that, in a good sampling condition and for a special class of nonsmooth functions, it is possible to move superlinearly to the solution.

One might view our method as a GS algorithm that incorporates some elements of Bundle Methods developed over the years [17, 33], but still keeps the GS facilities to handle nonconvex functions. This last characteristic is in agreement with Kiwiel's expectation [27]

> "We believe, however, that deeper understanding of their [GS and Bundle Methods] similarities and differences should lead to new variants."

In order to obtain rapid local convergence, the new algorithm needs to look at the VU-decomposition of the space [31, 36]. Roughly speaking, the method behaves like the cutting-plane methods [15, 24] in the V-space, whereas into the U-space (a smooth subspace for the objective function) it emulates the quasi-Newton techniques. For this purpose, we need not only to evaluate the gradients at the sample points, but also their respective function values. This procedure does not produce a significant increase in computational time, since, in most cases, the computational effort of evaluating the function value is fundamental in evaluating the gradient as well, so by computing the gradient one can obtain the function value essentially for free.

Finally, we believe that the results obtained in this chapter are a step further into the study of a practical algorithm with rapid local convergence to minimize nonsmooth and nonconvex functions. A future work assessing its performance in an extensive class of

nonsmooth functions is needed to determine how efficient the proposed algorithm is. For now, we limit ourselves to the convergence theory and the presentation of some illustrative examples.

4.1 Motivation and the new algorithm

As in the previous chapter, we will be interested in solving a minimax problem, but now considering $\mathcal{W} = \mathbb{R}^n$, i.e.,

$$\min_{x\in\mathbb{R}^n} \left(f(x) := \max_{1\leqslant i\leqslant p}\{\phi_i(x)\} \right), \tag{4.1}$$

where the functions $\phi_i : \mathbb{R}^n \to \mathbb{R}$ are all of class C^2, but they are not necessarily known. Again, we only ask that the function f may be represented as a maximum of functions, that is distinct from the case in which the functions that comprise f are known. For such a case, many studies have been developed (see [14] and references therein). Additionally, we also suppose the affinely independence presented in the previous chapter, but now we assume this condition for a larger set of points in the domain of f.

Assumption 4.1. *For all $x \in \mathbb{R}^n$ with $|\mathcal{I}(x)| \geqslant 2$, the gradients $\{\nabla\phi_i(x)\}_{i\in\mathcal{I}(x)}$ compose an affinely independent set, that is,*

$$\sum_{i\in\mathcal{I}(x)} \alpha_i\nabla\phi_i(x) = 0 \quad \textit{and} \quad \sum_{i\in\mathcal{I}(x)} \alpha_i = 0 \iff \alpha_i = 0, \ \forall i \in \mathcal{I}(x).$$

4.1.1 Motivational example

Suppose that we have $f(x) = |x| = \max\{x, -x\}$ and we want to start an iteration of Algorithm 1. If we have that

$$m = 2, \quad \epsilon_0 = 1, \quad \epsilon_{\text{opt}} < 1, \quad x_0 = 0.5, \quad x_{0,1} < 0 \ \text{ and } \ x_{0,2} > 0,$$

then $f'(x_{0,1}) = -1$, $f'(x_{0,2}) = 1$ and $\tilde{g}_0 = 0$ in Step 2. Consequently, by Step 3, we skip Steps 4 and 5 and go directly to Step 6, which starts a new iteration. Although this routine indicates that we have an ϵ_0-stationary point for f, this procedure does not allow us to move. Moreover, it prevents the algorithm to take an action when it has a complete information about the function, that is, when we have points sampled in the sets

$$X^- = \{x \in \mathbb{R} \mid x < 0\} \ \text{ and } \ X^+ = \{x \in \mathbb{R} \mid x > 0\}.$$

As a consequence, we see that the method only gets a chance to move when either x_k and the sampled points are all in X^- or all in X^+. Moreover, in this scenario, the GS method behaves exactly as the steepest descent method.

This undesirable behavior can be explained by the lack of information about the function values at the sampled points. Indeed, taking a careful look into the quadratic

optimization problem that is solved in Step 2, it is possible to see that its dual problem is given by

$$\begin{aligned} \min_{(d,z)} \quad & z + \frac{1}{2} d^T H_k d \\ \text{s.t.} \quad & \tilde{G}_k^T d \leqslant ze, \end{aligned}$$

where $z \in \mathbb{R}$ and d is the vector d_k that appears in Step 4. Equivalently, considering $x_{k,0} := x_k$, the same direction d_k can be obtained if we solve

$$\min_{d \in \mathbb{R}^n} \left\{ \max_{0 \leqslant i \leqslant m} \left\{ f(x_k) + \nabla f(x_{k,i})^T d \right\} + \frac{1}{2} d^T H_k d \right\}. \tag{4.2}$$

Notice, however, that if we use the function values of each sampled point instead of $f(x_k)$, i.e., if we solve

$$\min_{d \in \mathbb{R}^n} \left\{ \max_{1 \leqslant i \leqslant m} \left\{ f(x_{k,i}) + \nabla f(x_{k,i})^T (x_k - x_{k,i}) + \nabla f(x_{k,i})^T d \right\} + \frac{1}{2} d^T H_k d \right\}, \tag{4.3}$$

we would have a better model for the function f than the original one (closer to the cutting-plane method). Furthermore, the new quadratic optimization problem allows us to move when we have sampled in both "faces" of f, that is, in X^- and X^+. Lastly, observe that in (4.3), we do not use the objective function value at the current iterate x_k neither the gradient $\nabla f(x_k)$. As we shall see later, these omissions do not prevent the algorithm to converge and introduce an advantage over the GS method, since the Differentiability Check at Step 5 is no longer necessary.

Unfortunately, this new quadratic programming problem comes with a price: the vector d_k might not be a descent direction for f at x_k (especially under a bad sampling condition), a property that is always true if we solve (4.2). Therefore, to have an algorithm that uses the function values at all sampled points, we must overcome this issue.

4.1.2 New algorithm

In order to surpass the difficulty of not having a descent direction under a bad sample, we replace the Armijo's line search by a trust-region procedure. Besides, to have a smooth problem to solve, instead of dealing with (4.3), we solve at each iteration the following quadratic optimization problem

$$\begin{aligned} \min_{(d,z)} \quad & z + \frac{1}{2} d^T H_k d \\ \text{s.t.} \quad & \tilde{f}_k + G_k^T d \leqslant ze \\ & \|d\|_\infty \leqslant \Delta_k, \end{aligned} \tag{4.4}$$

where $\tilde{f}_k = [f(x_{k,1}) + \nabla f(x_{k,1})^T (x_k - x_{k,1}), \ldots, f(x_{k,m}) + \nabla f(x_{k,m})^T (x_k - x_{k,m})]^T$, $G_k = [\nabla f(x_{k,1}) \ \ldots \nabla f(x_{k,m})]$ and $\|d\|_\infty \leqslant \Delta_k$ stands for the trust-region constraint, for some

$\Delta_k > 0$. Consequently, its dual optimization problem, after changing variables, can be viewed as

$$\begin{aligned} \max_{(\lambda,\omega)\in\mathbb{R}^{m+n}} \quad & \lambda^T \tilde{f}_k - \frac{1}{2}(G_k\lambda + \omega)^T H_k^{-1}(G_k\lambda + \omega) - \Delta_k\|\omega\|_1 \\ \text{s.t.} \quad & \lambda^T e = 1 \\ & \lambda \geqslant 0. \end{aligned}$$

With these modifications in mind, we introduce the proposed algorithm (Algorithm 2), also referred to as GraFuS, which stands for Gradient and Function Sampling. Together with the exhibition of our new method, we must highlight some of its properties that will be important for the good understanding of the convergence results.

- The generated sequence of function values is monotone decreasing, i.e., $f(x_{k+1}) < f(x_k)$, for all $k \in \mathbb{N}$;
- the sequence $\{\nu_k\}$ is also monotone decreasing and it is a measure of how far we are from a stationary point as the sampling radius is related with the value of ν_k ;
- the role played by the exponent σ_l in the algorithm will be clarified at the local convergence section. Furthermore, its definition at Step 1 has the purpose of providing freedom for such a parameter, so that it may be modified any time the algorithm performs the referred step.

Notation Glossary	
k: outer iteration counter	ν_k: optimality measure
l: inner iteration counter	ν_{opt}: optimality certificate tolerance
x_k: current iterate	ι: exponent for updating ν_k
m: number of sampled points	$\epsilon_{k,l}$: related to the current sampling size
γ_Δ: constant related to the trust region	$\Delta_{k,l}$: current trust-region size
γ_ϵ: constant related to the sampling size	θ: reduction factor for $\epsilon_{k,l}$ and $\Delta_{k,l}$
ρ: parameter of step acceptance	σ_l: exponent related to the sampling size

4.2 Convergence

As in the last chapters, we also suppose that Assumptions 2.1 and 4.1 hold. Moreover, we also use $\mathcal{I}(x) = \{i \mid \phi_i(x) = f(x)\}$.

Remark 4.1. *Again it is worth pointing out that Assumption 4.1 can be viewed as a way to guarantee that, for any fixed $j \in \mathcal{I}(x)$, the set*

$$\{\nabla\phi_i(x) - \nabla\phi_j(x)\}_{i\in\mathcal{I}(x)\setminus\{j\}}$$

is linearly independent for all $x \in \mathbb{R}^n$ with $|\mathcal{I}(x)| \geqslant 2$. This association will be of great importance for both the global and the local convergence results.

Algorithm 2: A Superlinear Gradient and Function Sampling-based method (GraFuS).

Step 0. Set $k, l = 0$, $x_0 \in \mathbb{R}^n$, $m \in \mathbb{N}$ with $m \geqslant n+1$ and fixed real numbers $\gamma_\epsilon, \gamma_\Delta > 0$, $0 < \nu_0, \theta, \rho < 1$, $0 \leqslant \nu_{\text{opt}} < \nu_0$ and $\iota > 1$. Finally, define the initial sampling radius and the maximum step size as $\epsilon_{0,0} = \nu_0$ and $\Delta_{0,0} = \gamma_\Delta \nu_0$, respectively.

Step 1. Set σ_l as any real number in $[1, 2]$ and choose

$$\left\{x_{k,1}^l, \ldots, x_{k,m}^l\right\} \subset \mathcal{B}(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l})$$

with randomly, independently and uniformly sampled elements.

Step 2. Find $(d_{k,l}, z_{k,l})$ and $(\lambda_{k,l}, \omega_{k,l})$ that solve, respectively, (4.4) and its dual problem, where $H_k \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix.

Step 3. If $\nu_k \leqslant \nu_{\text{opt}}$, then STOP! Otherwise, compute

$$\text{Ared}_{k,l} := f(x_k) - f(x_k + d_{k,l})$$

and

$$\text{Pred}_{k,l} := \max_{1\leqslant i\leqslant m} \left\{f(x_{k,i}^l) + \nabla f(x_{k,i}^l)^T (x_k - x_{k,i}^l)\right\} - \left(z_{k,l} + \frac{1}{2} d_{k,l}^T H_k d_{k,l}\right).$$

Step 4. If $\text{Ared}_{k,l} \leqslant \rho \text{Pred}_{k,l}$, then a null step is performed by setting $\Delta_{k,l+1} = \theta \Delta_{k,l}$, $\epsilon_{k,l+1} = \theta \epsilon_{k,l}$, $l \leftarrow l + 1$ and going back to Step 1. Otherwise, a serious step is taken by setting $x_{k+1} = x_k + d_{k,l}$ and $\nu_{k+1} = \max\{\min\{\nu_k, \|H_k^{-1} G_{k,l} \lambda_{k,l}\|_\infty\}, \nu_k^\iota\}$.

Step 5. Set $\epsilon_{k+1,0} = \nu_{k+1}$, $\Delta_{k+1,0} = \gamma_\Delta \nu_{k+1}$, $k \leftarrow k + 1$, $l \leftarrow 0$ and go back to Step 1.

4.2.1 Global convergence

First, we present a technical lemma that guarantees that at most $n+1$ functions will assume the maximum of f at a fixed point $x \in \mathbb{R}^n$. In addition, we prove that, for each ϕ_j, with $j \in \mathcal{I}(x)$, there is a sufficiently small open set such that ϕ_j strictly assumes the maximum value at this specific set.

Lemma 4.1. *Let x be any point in $\mathbb{R}^n$ and j be any fixed index in $\mathcal{I}(x)$. Then, $|\mathcal{I}(x)| \leqslant n+1$. Moreover, there exists $r > 0$ such that for all $\epsilon \in (0, r)$, we can find a set $\mathcal{C}_j(x, \epsilon) \subset \mathcal{B}(x, \epsilon)$ with $\text{int}(\mathcal{C}_j(x, \epsilon)) \neq \varnothing$, for which $x \notin \mathcal{C}_j(x, \epsilon)$ and*

$$\phi_j(x^j) > \max_{\substack{1\leqslant i\leqslant p \\ i\neq j}} \phi_i(x^j), \quad \forall x^j \in \mathcal{C}_j(x, \epsilon).$$

Proof. Let us prove first that $|\mathcal{I}(x)| \leqslant n + 1$. If $|\mathcal{I}(x)| = 1$, the statement trivially holds. Therefore, we assume that $|\mathcal{I}(x)| \geqslant 2$. Besides, we suppose without any loss of generality

that $\mathcal{I}(x) = \{1, \ldots, r\}$. Then, let $\alpha_2, \ldots, \alpha_r \in \mathbb{R}$ be any real numbers such that

$$\sum_{i=2}^{r} \alpha_i \left(\nabla \phi_i(x) - \nabla \phi_1(x)\right) = 0.$$

Then, it follows that

$$-\left(\sum_{i=2}^{r} \alpha_i\right) \nabla \phi_1(x) + \sum_{i=2}^{r} \alpha_i \nabla \phi_i(x) = 0,$$

and by Assumption 4.1, we have $\alpha_2 = \ldots = \alpha_r = 0$. Consequently,

$$\mathcal{A} := \{\nabla \phi_i(x) - \nabla \phi_1(x)\}_{i \in \mathcal{I}(x) \setminus \{1\}}$$

forms a linearly independent set. So, $|\mathcal{A}| \leqslant n$, which implies that $|\mathcal{I}(x)| \leqslant n + 1$.

Now, for the other result, we also have that, if $|\mathcal{I}(x)| = 1$, then the proof is straightforward by a continuity argument. So, let us suppose that $|\mathcal{I}(x)| \geqslant 2$ and $\mathcal{I}(x) = \{1, \ldots, r\}$. By Assumption 4.1, given a fixed $s \in \mathcal{I}(x)$ and any $j \in \mathcal{I}(x)$ with $j \neq s$, we have that $v_j := \nabla \phi_j(x) - \nabla \phi_s(x)$ cannot be written as a linear combination of $\{v_i \mid i \in \mathcal{I}(x),\, i \neq j\}$ (to see this, just use the same arguments that we have used to prove $|\mathcal{I}(x)| \leqslant n + 1$ and notice that the set formed by the vectors v_j's is linearly independent). Thus, it is possible to find a unitary $d_j \in \mathbb{R}^n$ such that $v_j^T d_j > 0$ and

$$v_i^T d_j = 0, \;\; i \neq j \;\; \text{with} \;\; i \in \mathcal{I}(x).^{1}$$

Consequently, it follows that $\nabla \phi_j(x)^T d_j > \nabla \phi_s(x)^T d_j$ and

$$\nabla \phi_i(x)^T d_j = \nabla \phi_s(x)^T d_j, \;\; i \neq j \;\; \text{with} \;\; i \in \mathcal{I}(x).$$

So, since $\phi_i \in C^2$, for all $i \in \mathcal{I}(x)$, we have that for all fixed $w_j \in \mathbb{R}^n$ it follows that

$$\begin{aligned}
\phi_i(x + \epsilon(d_j + w_j)) &= \phi_i(x) + \epsilon \nabla \phi_i(x)^T (d_j + w_j) + O(\epsilon^2), \;\; i \in \mathcal{I}(x), \;\; i \neq j, \\
\phi_j(x + \epsilon(d_j + w_j)) &= \phi_j(x) + \epsilon \nabla \phi_j(x)^T (d_j + w_j) + O(\epsilon^2).
\end{aligned}$$

Now, subtracting the first equation above from the second one and dividing the result by ϵ, we obtain, for all $i \in \mathcal{I}(x)$ with $i \neq j$, that

$$\begin{aligned}
\frac{\phi_j(x + \epsilon(d_j + w_j)) - \phi_i(x + \epsilon(d_j + w_j))}{\epsilon} &= \nabla \phi_j(x)^T (d_j + w_j) \\
&\quad - \nabla \phi_i(x)^T (d_j + w_j) + O(\epsilon).
\end{aligned}$$

Consequently, supposing that

$$w_j \in \mathcal{B}(0, \delta) \subset \mathbb{R}^n, \text{ where } \delta := \min_{\substack{i \in \mathcal{I}(x) \\ i \neq j}} \left\{ \frac{[\nabla \phi_j(x) - \nabla \phi_i(x)]^T d_j}{2 \|\nabla \phi_j(x) - \nabla \phi_i(x)\|} \right\} > 0,$$

[1] For example, setting s_j as the orthogonal projection of v_j over the hyperplane generated by $\{v_i \mid i \in \mathcal{I}(x),\, i \neq j\}$, one can consider $d_j = (v_j - s_j)/\|v_j - s_j\|$.

we must have, for all $i \in \mathcal{I}(x)$ with $i \neq j$, that

$$\begin{aligned}\frac{\phi_j(x+\epsilon(d_j+w_j))-\phi_i(x+\epsilon(d_j+w_j))}{\epsilon} &= [\nabla\phi_j(x)-\nabla\phi_i(x)]^T d_j \\ &\quad + [\nabla\phi_j(x)-\nabla\phi_i(x)]^T w_j + O(\epsilon) \\ &\geqslant [\nabla\phi_j(x)-\nabla\phi_i(x)]^T d_j \\ &\quad - \|\nabla\phi_j(x)-\nabla\phi_i(x)\|\|w_j\| + O(\epsilon) \\ &> \frac{[\nabla\phi_j(x)-\nabla\phi_i(x)]^T d_j}{2} + O(\epsilon).\end{aligned}$$

From the inequality above and noticing that $[\nabla\phi_j(x)-\nabla\phi_i(x)]^T d_j > 0$, for all $i \in \mathcal{I}(x)$ with $i \neq j$, it is possible to find $r_j > 0$ small enough such that for all $\epsilon \in (0, r_j)$ the following relation holds

$$\phi_j(x+\epsilon(d_j+w_j)) > \phi_i(x+\epsilon(d_j+w_j)), \;\; i \in \mathcal{I}(x), \;\; i \neq j.$$

To complete the proof, notice that the functions ϕ_i are continuous, and therefore, it is possible to find $\tilde{r} > 0$ such that for all $y \in \mathcal{B}(x, \tilde{r})$ the following holds

$$\phi_a(y) > \phi_b(y), \;\; a \in \mathcal{I}(x), \;\; b \notin \mathcal{I}(x).$$

So, setting $r := \min\{r_1, \ldots, r_p, \tilde{r}\}$ and choosing $\epsilon \in (0, r)$, we have that the set

$$\mathcal{C}_j(x,\epsilon) := \{x + \tau(d_j + w_j) \mid 0 < \tau < \epsilon/2, \;\; w_j \in \mathcal{B}(0,\delta), \;\; j \in \mathcal{I}(x)\},$$

satisfies the properties previously claimed. □

From the above result, we can see that for any $\epsilon > 0$ (even when $\epsilon \geqslant r$, since in this case we have $\mathcal{B}(x,r) \subset \mathcal{B}(x,\epsilon)$), the following set is not empty

$$\mathcal{S}_j(x,\epsilon) := \operatorname{int}\left\{ y \in \mathcal{B}(x,\epsilon) \;\middle|\; \phi_j(y) > \max_{\substack{1\leqslant i\leqslant p \\ i\neq j}} \phi_i(y) \right\}, \;\; j \in \mathcal{I}(x). \tag{4.5}$$

So, we can proceed with two additional results. They guarantee that GraFuS is well defined, i.e., the algorithm will not cycle forever from Step 4 to Step 1. Specifically, the first result tells us that under a good set of sampled points, it is possible to obtain Ared $> \rho$Pred at Step 4 (the proof of the result is based on ideas from [51]).

Lemma 4.2. *In Algorithm 2, consider fixed outer and inner iterations, denoted by k and l, respectively. Let $\overline{x} \in \mathbb{R}^n$ be a nonstationary point for the function $f : \mathbb{R}^n \to \mathbb{R}$, $\rho \in (0,1)$ be a fixed real number and $\mathcal{S}_j(\overline{x},\epsilon)$ be the set defined in (4.5) for any $\epsilon > 0$. Therefore, there exist $\overline{\Delta}$ and $\overline{\delta} > 0$ such that, if the following hypotheses hold*

i) $x_k \in \mathcal{B}(\overline{x}, \overline{\delta})$;

ii) $0 < \Delta_{k,l} < \overline{\Delta}$;

iii) there exist $\overline{\epsilon} \equiv \overline{\epsilon}(k,l) > 0$ *and* $M > 0$ *such that*

a) for all $j \in \mathcal{I}(\overline{x})$, *we have* $\mathcal{S}_j(\overline{x}, \overline{\epsilon}) \subset \mathcal{B}(x_k, M \cdot \Delta_{k,l})$;

b) for all $j \in \mathcal{I}(\overline{x})$, *there exists* $x_{k,i}^l \in \mathcal{S}_j(\overline{x}, \overline{\epsilon})$, *for some* $i \in \{1, \ldots, m\}$;

c) for all $i \in \{1, \ldots, m\}$, *it is possible to find* $j \in \mathcal{I}(\overline{x})$ *such that* $x_{k,i}^l \in \mathcal{S}_j(\overline{x}, \overline{\epsilon})$,

then

$$Ared_{k,l} > \rho Pred_{k,l}.$$

Proof. First, we choose $r > 0$ as a sufficiently small number such that for all $x \in \mathcal{B}(\overline{x}, r)$, we have

$$\phi_j(x) > \max_{\substack{1 \leqslant i \leqslant p \\ i \notin \mathcal{I}(\overline{x})}} \phi_i(x), \text{ for all } j \in \mathcal{I}(\overline{x}).$$

Since $\overline{x}$ is not a stationary point for f, we must have that $0 \notin \overline{\partial} f(\overline{x})$. Recalling that $\overline{\partial} f(\overline{x})$ is a closed and convex set, it follows by the Hyperplane Separation Theorem [3, Section 2.5] that there exist a unitary vector $v \in \mathbb{R}^n$ and a scalar $\tau > 0$ such that

$$s^T v \leqslant -\tau, \ \ \forall s \in \overline{\partial} f(\overline{x}).$$

Since the generalized directional derivative of f at $\overline{x}$ in the direction v is given by

$$f^\circ(\overline{x}; v) = \limsup_{\substack{x \to \overline{x} \\ t \downarrow 0}} \frac{f(x + tv) - f(x)}{t} = \max\{s^T v \ : \ s \in \overline{\partial} f(\overline{x})\},$$

we have that $f^\circ(\overline{x}; v) \leqslant -\tau$. Thus, there exist $\overline{\Delta} \in (0, r)$ and $\overline{\delta} \in (0, r)$ such that for all $x \in \mathcal{B}\left(\overline{x}, \overline{\delta}\right)$ and $\Delta \in (0, \overline{\Delta})$, we have

$$f(x + \Delta v) - f(x) < -\frac{\tau}{2}\Delta. \tag{4.6}$$

Now, let us keep this information in mind and proceed with a parallel idea. Let us suppose that the hypotheses *i*), *ii*) and *iii*) hold for $\overline{\delta}$ and $\overline{\Delta}$ found above. Then

$$\begin{aligned} f(x_k) &= \max_{i \in \mathcal{I}(\overline{x})} \{\phi_i(x_k)\} \\ &= \max_{1 \leqslant i \leqslant m} \{f(x_{k,i}^l) + \nabla f(x_{k,i}^l)^T (x_k - x_{k,i}^l)\} + o(\Delta_{k,l}) \\ &\quad \text{(notice that } x_{k,i}^l \in \mathcal{B}(x_k, M \cdot \Delta_{k,l})) \end{aligned}$$

and

$$\begin{aligned} f(x_k + d_{k,l}) &= \max_{i \in \mathcal{I}(\overline{x})} \{\phi_i(x_k + d_{k,l})\} \\ &= \max_{1 \leqslant i \leqslant m} \{f(x_{k,i}^l) + \nabla f(x_{k,i}^l)^T (x_k + d_{k,l} - x_{k,i}^l)\} + o(\Delta_{k,l}) \\ &\quad \text{(notice that } x_{k,i}^l \in \mathcal{B}(x_k, M \cdot \Delta_{k,l}) \text{ and that } \|d_{k,l}\|_\infty \leqslant \Delta_{k,l}). \end{aligned}$$

So, we have $\text{Ared}_{k,l} = f(x_k) - f(x_k + d_{k,l}) = \text{Pred}_{k,l} + o(\Delta_{k,l})$. Consequently, to prove the statement, we just need to show that $\Delta_{k,l} = O(\text{Pred}_{k,l})$, since we would have, for any $\eta = (1-\rho) \in (0,1)$, a sufficiently small $\overline{\Delta} > 0$ such that

$$\text{Ared}_{k,l} - \text{Pred}_{k,l} = o(\Delta_{k,l}) > -\eta \text{Pred}_{k,l},$$

which yields that $\text{Ared}_{k,l} > (1-\eta)\text{Pred}_{k,l} = \rho \text{Pred}_{k,l}$. So, to show that such a condition holds, we define

$$\hat{z} = \max_{1 \leqslant i \leqslant m} \{f(x_{k,i}^l) + \nabla f(x_{k,i}^l)^T (x_k + \Delta_{k,l} v - x_{k,i}^l)\}.$$

Therefore, since $(d_{k,l}, z_{k,l})$ is the solution of the quadratic programming problem at Step 2, we have that $z_{k,l} \leqslant \hat{z} + o(\Delta_{k,l})$, and hence,

$$\text{Pred}_{k,l} \geqslant \max_i \{f(x_{k,i}^l) + \nabla f(x_{k,i}^l)^T (x_k - x_{k,i}^l)\} - \left(\hat{z} + \frac{\Delta_{k,l}^2}{2} v^T H_k v\right) + o(\Delta_{k,l}).$$

Consequently, it yields that

$$\begin{aligned} \text{Pred}_{k,l} &\geqslant f(x_k) - f(x_k + \Delta_{k,l} v) + o(\Delta_{k,l}) \\ &> \frac{\tau}{2} \Delta_{k,l} + o(\Delta_{k,l}), \end{aligned}$$

where the last inequality comes from (4.6). Therefore, if $\overline{\Delta}$ is small enough, we obtain the desired result. □

With the above result, we present the following lemma, which claims that if GraFuS is at iteration k and x_k is not a stationary point for f, then the index l of the inner iteration has an upper limit (with probability one).

Lemma 4.3. *Suppose that for an iteration k, we have that x_k is not a stationary point for f. Then, with probability one, there exists $\bar{l} \in \mathbb{N}$ such that the indices of the inner iterations satisfy $l \leqslant \bar{l}$.*

Proof. Let us assume, by contradiction, that such $\bar{l}$ does not exist, i.e., $l \to \infty$ at the iteration k. Consequently, we must have that $\text{Ared}_{k,l} \leqslant \rho \text{Pred}_{k,l}$, for all $l \in \mathbb{N}$. Additionally, by the way we have designed our algorithm, we see that

$$\epsilon_{k,l} = \frac{1}{\gamma_\Delta} \Delta_{k,l}, \quad \text{for all } k, l \in \mathbb{N},$$

and by the contradiction hypothesis the following holds: $\Delta_{k,l} \to 0$ as $l \to \infty$.

Therefore, setting $\overline{x} := x_k$ in Lemma 4.2, it is straightforward to see that at some $\tilde{n} \in \mathbb{N}$, if $l \geqslant \tilde{n}$, then hypotheses *i*) and *ii*) of Lemma 4.2 are valid. Moreover, considering $\bar{\epsilon} := \gamma_\epsilon \epsilon_{k,l}^{\sigma_l}$ and $M := \gamma_\epsilon \max\{\gamma_\Delta^{-1}, \gamma_\Delta^{-2}\}$ for a fixed inner iteration l, we will satisfy hypothesis *iii*) item *a*) of Lemma 4.2. Therefore, if at this specific inner iteration l

we do not have $\text{Ared}_{k,l} > \rho\text{Pred}_{k,l}$, it is due to the fact that we did not sample the points properly, i.e, the items b) and/or c) of hypothesis iii) were not fulfilled. So, since $l \to \infty$ by the contradiction hypothesis we have made, it is also true that the next inner iteration will not satisfy items b) and/or c) and so on. We claim that this behavior has probability zero to occur.

Indeed, let us assume a fixed $j \in \mathcal{I}(x_k)$ and notice that by the way we have defined d_j and $\mathcal{C}_j(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l})$ in the proof of Lemma 4.1, we have that (for $\gamma_\epsilon \epsilon_{k,l}^{\sigma_l}$ sufficiently small) $\mathcal{B}_j^{k,l} \subset \mathcal{C}_j(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l})$, where

$$\mathcal{B}_j^{k,l} := \mathcal{B}\left(x_k + \frac{\gamma_\epsilon \epsilon_{k,l}^{\sigma_l}}{4} d_j, \frac{\gamma_\epsilon \epsilon_{k,l}^{\sigma_l}}{8} \min_{\substack{i \in \mathcal{I}(x_k) \\ i \neq j}} \left\{ \frac{[\nabla \phi_j(x_k) - \nabla \phi_i(x_k)]^T d_j}{2\|\nabla \phi_j(x_k) - \nabla \phi_i(x_k)\|} \right\} \right).$$

Consequently, the volume of $\mathcal{B}_j^{k,l}$ in $\mathbb{R}^n$ is given by

$$\text{Vol}\left(\mathcal{B}_j^{k,l}\right) = \frac{\pi^{n/2}}{\Gamma(n/2+1)} \left(\min_{\substack{i \in \mathcal{I}(x_k) \\ i \neq j}} \left\{ \frac{[\nabla \phi_j(x_k) - \nabla \phi_i(x_k)]^T d_j}{2\|\nabla \phi_j(x_k) - \nabla \phi_i(x_k)\|} \right\} \right)^n \left(\frac{\gamma_\epsilon \epsilon_{k,l}^{\sigma_l}}{8} \right)^n,$$

where Γ is the Gamma function [23]. On the other hand, it follows that

$$\text{Vol}(\mathcal{B}(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l})) = \frac{\pi^{n/2}}{\Gamma(n/2+1)} \left(\gamma_\epsilon \epsilon_{k,l}^{\sigma_l}\right)^n.$$

Therefore, since the sampled points are chosen in $\mathcal{B}(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l})$ and

$$\mathcal{B}_j^{k,l} \subset \mathcal{C}_j(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l}) \subset \mathcal{S}_j(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l}),$$

we must have, for all $i \in \{1, \ldots, m\}$, that the conditional probability

$$\text{Proj}(x_{k,i}^l \in \mathcal{S}_j(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l}) \mid x_{k,i}^l \in \mathcal{B}(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l})) = \frac{\text{Vol}(\mathcal{S}_j(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l}))}{\text{Vol}(\mathcal{B}(x_k, \gamma_\epsilon \epsilon_{k,l}^{\sigma_l}))}$$

must be greater than the following strictly positive number

$$\frac{1}{8^n} \left(\min_{\substack{i \in \mathcal{I}(x_k) \\ i \neq j}} \left\{ \frac{[\nabla \phi_j(x_k) - \nabla \phi_i(x_k)]^T d_j}{2\|\nabla \phi_j(x_k) - \nabla \phi_i(x_k)\|} \right\} \right)^n.$$

With this inequality, we conclude that the probability of the items b) and c) of hypothesis iii) to happen together is strictly positive and does not depend on l. Therefore, the probability of $l \to \infty$ is zero, which concludes the proof. □

Finally, we are close to reach the convergence theorem of GraFuS. For that goal, we only need to prove a last technical lemma. Furthermore, to have a clearer proof, from now on we will denote by $\bar{l}_k$ the largest value of the index l at the iteration k.

Lemma 4.4. *Let us consider the GraFuS algorithm. If* $\text{Pred}_{k,\bar{l}_k}/\Delta_{k,\bar{l}_k} \to 0$ *as* $k \to \infty$, *then* $\|G_{k,\bar{l}_k} \lambda_{k,\bar{l}_k}\| \to 0$.

Proof. First, notice that the quadratic programming problem presented in (4.4) satisfies the Slater's condition. Indeed, if one considers $d_k = 0$ and $z_k = \max\{\tilde{f}_k\} + 1$ in (4.4), then we see that all inequalities are strictly satisfied. Thus, since the problem is also convex, we can guarantee that the quadratic programming problem satisfies strong duality. So, we have

$$\begin{aligned} z_{k,\bar{l}_k} + \frac{1}{2} d_{k,\bar{l}_k}^T H_k d_{k,\bar{l}_k} = \; & \lambda_{k,\bar{l}_k}^T \tilde{f}_{k,\bar{l}_k} \\ & - \frac{1}{2} \left(G_{k,\bar{l}_k} \lambda_{k,\bar{l}_k} + \omega_{k,\bar{l}_k} \right)^T H_k^{-1} \left(G_{k,\bar{l}_k} \lambda_{k,\bar{l}_k} + \omega_{k,\bar{l}_k} \right) \\ & - \Delta_{k,\bar{l}_k} \|\omega_{k,\bar{l}_k}\|_1. \end{aligned}$$

Thus, defining

$$\alpha_k := \frac{1}{2} \left(G_{k,\bar{l}_k} \lambda_{k,\bar{l}_k} + \omega_{k,\bar{l}_k} \right)^T H_k^{-1} \left(G_{k,\bar{l}_k} \lambda_{k,\bar{l}_k} + \omega_{k,\bar{l}_k} \right) + \Delta_{k,\bar{l}_k} \|\omega_{k,\bar{l}_k}\|_1, \tag{4.7}$$

it yields

$$\begin{aligned} \lambda_{k,\bar{l}_k}^T \tilde{f}_{k,\bar{l}_k} - \alpha_k = z_{k,\bar{l}_k} + \frac{1}{2} d_{k,\bar{l}_k}^T H_k d_{k,\bar{l}_k} \Rightarrow \alpha_k &= \lambda_{k,\bar{l}_k}^T \tilde{f}_{k,\bar{l}_k} \\ & - \left(z_{k,\bar{l}_k} + \frac{1}{2} d_{k,\bar{l}_k}^T H_k d_{k,\bar{l}_k} \right) \\ \Rightarrow \alpha_k &\leqslant \mathrm{Pred}_{k,\bar{l}_k} \\ & \text{(since } \lambda_{k,\bar{l}_k} \geqslant 0 \text{ and } e^T \lambda_{k,\bar{l}_k} = 1) \\ \Rightarrow \frac{\alpha_k}{\Delta_{k,\bar{l}_k}} &\leqslant \frac{\mathrm{Pred}_{k,\bar{l}_k}}{\Delta_{k,\bar{l}_k}} \\ \Rightarrow \frac{\alpha_k}{\Delta_{k,\bar{l}_k}} &\to 0. \end{aligned}$$

Consequently, by Assumption 2.1 and (4.7), we obtain $\|G_{k,\bar{l}_k} \lambda_{k,\bar{l}_k}\| \to 0$. □

Now, we reach the main goal of this subsection. Below, we prove the global convergence (with probability one) of the proposed algorithm.

Theorem 4.1. *Suppose that the GraFuS algorithm produces a bounded sequence of points* $\{x_k\}$ *with* $\nu_{opt} = 0$. *Then, with probability one, there is a cluster point* $\overline{x}$ *of this sequence which is a stationary point for* f.

Proof. We split the proof in two complementary cases:

i) There are an infinite set of indices $\mathcal{K}_1 \subset \mathbb{N}$ and a real number $\bar{\epsilon} > 0$ such that $\epsilon_{k,\bar{l}_k} \geqslant \bar{\epsilon}$ for all $k \in \mathcal{K}_1$.

ii) The sampling radius along the iterations satisfy $\epsilon_{k,\bar{l}_k} \underset{k \in \mathbb{N}}{\to} 0$.

Initially, let us suppose that case i) holds. So, noticing that $\epsilon_{k,\bar{l}_k} \leqslant \nu_k$, for all $k \in \mathbb{N}$, and that $\{\nu_k\}$ is a monotone decreasing sequence, we see clearly that there must exist $\overline{\nu}$ such that $\nu_k \geqslant \overline{\nu}$, for all $k \in \mathbb{N}$. Additionally, we claim that there exists $\mu > 0$ such that $\Delta_{k,\bar{l}_k}\mu \leqslant \text{Pred}_{k,\bar{l}_k}$, for all $k \in \mathbb{N}$. Indeed, if this statement were false, there would exist an infinite set of indices $\tilde{\mathcal{K}}$ such that

$$\text{Pred}_{k,\bar{l}_k}/\Delta_{k,\bar{l}_k} \underset{k\in\tilde{\mathcal{K}}}{\to} 0.$$

However, by Lemma 4.4, it would yield that

$$\|G_{k,\bar{l}_k}\lambda_{k,\bar{l}_k}\| \underset{k\in\tilde{\mathcal{K}}}{\to} 0.$$

Therefore, we would have $\nu_k \to 0$, and consequently, that $\epsilon_{k,\bar{l}_k} \to 0$, which is a contradiction with case i). Thus, there must exist $\mu > 0$ such that $\Delta_{k,\bar{l}_k}\mu \leqslant \text{Pred}_{k,\bar{l}_k}$, for all $k \in \mathbb{N}$. Moreover, since

$$\epsilon_{k,l} = \frac{1}{\gamma_\Delta}\Delta_{k,l}, \text{ for all } k, l \in \mathbb{N},$$

we see that $\Delta_{k,\bar{l}_k} \geqslant \gamma_\Delta\overline{\epsilon}$, for all $k \in \mathcal{K}_1$. Consequently,

$$\text{Ared}_{k,\bar{l}_k} > \rho\text{Pred}_{k,\bar{l}_k}, \text{ for all } k \in \mathcal{K}_1 \Rightarrow f(x_k) - f(x_{k+1}) > \rho\mu\gamma_\Delta\overline{\epsilon}, \text{ for all } k \in \mathcal{K}_1. \quad (4.8)$$

Now, since $\{x_k\}$ is a bounded sequence by assumption, there must exist an infinite set of indices $\mathcal{K}_2 \subset \mathcal{K}_1$ such that

$$x_k \underset{k\in\mathcal{K}_2}{\to} \hat{x}, \text{ for some } \hat{x} \in \mathbb{R}^n.$$

So, considering $s_{\mathcal{K}_2}(k)$ as the index in $\mathcal{K}_2$ that comes right after $k \in \mathcal{K}_2$, we have

$$\sum_{k\in\mathcal{K}_2}(f(x_k) - f(x_{k+1})) \leqslant \sum_{k\in\mathcal{K}_2}\left(f(x_k) - f\left(x_{s_{\mathcal{K}_2}(k)}\right)\right) = f(x_w) - f(\hat{x}) < \infty,$$

with $w \in \mathbb{N}$ being the first index in $\mathcal{K}_2$. However, this is a condition that goes against (4.8). Therefore, the case i) is an impossible event and we must consider case ii).

Suppose that case ii) holds. Since $\{x_k\}$ is bounded, there exists at least one cluster point $\overline{x}$ of this sequence. Hence, there is $\mathcal{K} \subset \mathbb{N}$ such that

$$x_k \underset{k\in\mathcal{K}}{\to} \overline{x}.$$

Now, let us add two additional hypotheses to case ii):

a) The point $\overline{x}$ is not a stationary point for f;

b) There exists $M > 0$ such that $\nu_k > M$, for all $k \in \mathbb{N}$.

Then, we choose $\overline{\delta}, \overline{\Delta} > 0$ as presented in Lemma 4.2 for the point $\overline{x}$. Since $\nu_k > M$, for all $k \in \mathbb{N}$, and $\epsilon_{k,\bar{l}_k} \to 0$ as $k \to \infty$, we have that, by the way we have designed GraFuS, $\epsilon_{k,\bar{l}_k}$ just keeps going smaller because $\bar{l}_k \to \infty$. As a consequence, it yields that there exist $k', l' \in \mathbb{N}$ such that for all $k \geqslant k'$ we have

$$\Delta_{k,l'} = \tilde{\Delta} := \left(\theta^{l'}\right) \gamma_\Delta \nu_k < \overline{\Delta} \quad \text{and} \quad \epsilon_{k,l'} = \tilde{\epsilon} := \left(\theta^{l'}\right) \nu_k = \frac{1}{\gamma_\Delta}\tilde{\Delta}.$$

Moreover, since $\overline{x}$ is a cluster point for the iteration sequence, we can find $\hat{k} \geqslant k'$ such that for all $k \geqslant \hat{k}$ and $k \in \mathcal{K}$, we have $x_k \in \mathcal{B}(\overline{x}, \min\{\gamma_\epsilon \tilde{\epsilon}, \overline{\delta}\}/4)$. So, for all $j \in \mathcal{I}(\overline{x})$ we have

$$x_k \in \mathcal{B}(\overline{x}, \min\{\gamma_\epsilon \tilde{\epsilon}^2, \overline{\delta}\}/4) \text{ and } \mathcal{S}_j(\overline{x}, \min\{\gamma_\epsilon \tilde{\epsilon}, \overline{\delta}\}/4) \subset \mathcal{B}(x_k, (\gamma_\epsilon/\gamma_\Delta)\tilde{\Delta}).$$

Therefore, the hypotheses *i*), *ii*) and *iii*) item *a*) of Lemma 4.2 are all satisfied. So, since $\bar{l}_k \to \infty$, we must have that items *b*) and/or *c*) of hypothesis *iii*) are not satisfied for every $k \geqslant \hat{k}$ and $l = l'$. However, this is an event with probability zero to happen, since the sets $\mathcal{S}_j(\overline{x}, \min\{\gamma_\epsilon \tilde{\epsilon}^{l'}, \overline{\delta}\}/4)$ are open and not empty. As a consequence, with probability one, at least one of the two possible situations below must happen:

a') The cluster point $\overline{x}$ is a stationary point for f;

b') There is no $M > 0$ such that $\nu_k > M$, for all $k \in \mathbb{N}$. In other words, $\nu_k \to 0$.

If a') holds the statement is proven. However, if only b') is valid, then there exist an infinite set of indices $\overline{\mathcal{K}} \subset \mathbb{N}$ and a sequence of vectors $\{v_k\} \subset \mathbb{R}^n$ such that

$$v_k \in \overline{\partial}_{\epsilon_{k,0}} f(x_k) \text{ and } \|v_k\| \to 0, \text{ for all } k \in \overline{\mathcal{K}}.$$

Thus, since $\{x_k\}$ is a bounded sequence, we can assume without loss of generality that

$$x_k \underset{k \in \overline{\mathcal{K}}}{\to} \tilde{x}, \text{ for some } \tilde{x} \in \mathbb{R}^n.$$

Hence, remembering that $\epsilon_{k,0} \to 0$ (since $\nu_k \to 0$), we have the desired result (see item *iii*) of [27, Lemma 3.2]), i.e., $0 \in \overline{\partial} f(\tilde{x})$ with probability one. □

We have proved that our proposed algorithm has at least one cluster point that is stationary for f. For that, we needed to assume that the method has generated a bounded sequence of iterations, which can easily be obtained by supposing that the function f has bounded level sets. In addition, we have shown that we do not need to know the functions that comprise f to converge. In fact, we have traded this knowledge by the chance of having a good set of sample points.

In the next subsection, we have the intent to show that, under a good sampling, it is possible to move superlinearly to a local minimizer of f. For such a goal, our analysis will involve the concept of U and V spaces.

4.2.2 Local convergence

In this subsection our efforts will be focused in enlightening the role played by the quadratic programming problem (4.4). In fact, under special circumstances, it is possible to see this quadratic problem as a local approximation of a new optimization problem that involves the smooth functions ϕ_i. Upon this new perspective, we can analyze the local convergence of the proposed method and obtain interesting results. However, since our method has a random nature and a good local information about the function is restricted to a good set of sampled points, it is reasonable to think that a good rate of convergence will not be achieved at every iteration. Therefore, the results presented here will be sustained on hypotheses that guarantee a good sampling.

To accomplish the aim of this subsection, we start supposing that $x_* \in \mathbb{R}^n$ is a local minimizer of the optimization problem presented in (4.1). Also, assume that $\mathcal{I}(x_*) = \{1, \ldots, r+1\}$, for some $r \leqslant n$. Therefore, consider any $x_k \in \mathbb{R}^n$ and the sampled points $x_{k,1}^{\bar{l}_k}, \ldots, x_{k,m}^{\bar{l}_k} \in \mathcal{B}(x_k, \gamma_\epsilon \epsilon_{k,\bar{l}_k}^{\sigma_{\bar{l}_k}})$. So, we admit the following hypotheses on problem (4.4):

H1) We have a good set of sampled points: for any $j \in \mathcal{I}(x_*)$ there is $i_j \in \{1, \ldots, m\}$ such that

$$\phi_j\left(x_{k,i_j}^{\bar{l}_k}\right) > \phi_s\left(x_{k,i_j}^{\bar{l}_k}\right), \quad \text{for all } s \in \{1, \ldots, p\}, \quad s \neq j. \tag{4.9}$$

For an easier exposition of our ideas, we will write without loss of generality that

$$\phi_i\left(x_{k,i}^{\bar{l}_k}\right) > \phi_s\left(x_{k,i}^{\bar{l}_k}\right), \quad \text{for all } s \in \{1, \ldots, p\}, \quad s \neq i \text{ and } i \in \mathcal{I}(x_*), \tag{4.10}$$

since by a simple rearrangement of the sampled points in (4.9) the inequality (4.10) holds. Moreover, notice that this condition is the same hypothesis made in H_ϕ in the last chapter;

H2) The first $r+1$ constraints are active at the solution;

H3) Only the first $r+1$ constraints are active at the solution[2].

Remark 4.2. *Notice that assuming H3, we are implicitly asking that the trust-region constraint is not active, an assumption that seems inevitable because of the random nature of the algorithm.*

Under those hypotheses, one can rewrite (4.4) as the following optimization problem

$$\begin{aligned} \min_{(d,z)\in\mathbb{R}^{n+1}} \quad & z + \frac{1}{2} d^T H_k d \\ \text{s.t.} \quad & \phi_i\left(x_{k,i}^{\bar{l}_k}\right) + \nabla\phi_i\left(x_{k,i}^{\bar{l}_k}\right)^T \left(x_k + d - x_{k,i}^{\bar{l}_k}\right) = z, \quad 1 \leqslant i \leqslant r+1. \end{aligned} \tag{4.11}$$

[2] We believe that this hypothesis may seem unnatural at first sight. For this reason, we have treated it in the Appendix.

Alternatively, it can also be viewed as

$$\begin{aligned} &\min_{d\in\mathbb{R}^n} \ \phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) + \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T\left(x_k + d - x_{k,r+1}^{\bar{l}_k}\right) + \frac{1}{2}d^T H_k d \\ &\text{s.t.} \ \ \tilde{\Phi}_k + \tilde{J}_k d = 0, \end{aligned} \tag{4.12}$$

where $\tilde{\Phi}_k \in \mathbb{R}^r$ with

$$\begin{aligned} (\tilde{\Phi}_k)_i := \ &\phi_i\left(x_{k,i}^{\bar{l}_k}\right) + \nabla\phi_i\left(x_{k,i}^{\bar{l}_k}\right)^T\left(x_k - x_{k,i}^{\bar{l}_k}\right) \\ &- \left[\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) + \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T\left(x_k - x_{k,r+1}^{\bar{l}_k}\right)\right], \ i \in \{1,\dots,r\}, \end{aligned}$$

and

$$\tilde{J}_k := \begin{pmatrix} \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \\ \vdots \\ \nabla\phi_r\left(x_{k,r}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \end{pmatrix}.$$

So, the minimization problem (4.4) can be viewed as a quadratic approximation of

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \ \phi_{r+1}(x) \\ &\text{s.t.} \ \ \Phi(x) = 0, \end{aligned} \tag{4.13}$$

where

$$\Phi(x) := \begin{pmatrix} \phi_1(x) - \phi_{r+1}(x) \\ \vdots \\ \phi_r(x) - \phi_{r+1}(x) \end{pmatrix}.$$

Moreover, it is straightforward to see that x_* is also a local minimizer for (4.13).

Notice that for any $s \in U(x)$, we have that f behaves smoothly along s at x, since the directional derivatives (considering s) of ϕ_i are all the same for $i \in \mathcal{I}(x)$. Consequently, the kernel of the Jacobian of $\Phi(x)$ will be of great importance to us, because it recovers the smooth subspace of f at x_* when x approaches x_*. Therefore, we denote by J_x the Jacobian of $\Phi(x)$ and by $Z_x^{\lhd}$ the matrix whose columns form a basis for the kernel of J_x. Moreover, from now on, our analysis will be restricted to the case that $r \in \{1,\dots,n-1\}$. The cases $r = 0$ and $r = n$ will be treated later (see Remark 4.4).

In light of Remark 4.1, due to the Assumption 4.1, it is possible to see that the map $J_x : \mathbb{R}^n \to \mathbb{R}^r$ is surjective for all x in a small neighborhood $\mathcal{W}$ of x_*. Hence, for $x \in \mathcal{W}$, there must exist $J_x^{\lhd} \in \mathbb{R}^{n\times r}$ such that $J_x J_x^{\lhd} = I_r$. Moreover, by [2, Lemma 14.3], one can see that there exists only one map

$$\begin{aligned} Z : \mathbb{R}^n &\longrightarrow \mathbb{R}^{(n-r)\times n} \\ x &\longmapsto Z_x \end{aligned}$$

such that $Z_x J_x^{\lhd}$ is a null matrix, $Z_x Z_x^{\lhd} = I_{n-r}$ and the following relation holds

$$Z_x^{\lhd} Z_x + J_x^{\lhd} J_x = I_n. \tag{4.14}$$

So, we may divide $\mathbb{R}^n$ into two subspaces, generated by the columns of $Z_x^{\triangleleft}$ and $J_x^{\triangleleft}$, respectively.

Now, coming back to the optimization problem (4.13), we define its Lagrangian function $\mathcal{L}(x,\lambda):\mathbb{R}^n\times\mathbb{R}^r\to\mathbb{R}$ as

$$\mathcal{L}(x,\lambda)=\phi_{r+1}(x)+\lambda^T\Phi(x). \tag{4.15}$$

By Remark 4.1, the feasible set of problem (4.13) satisfies the linear independence constraint qualification and thus there is only one $\lambda_*\in\mathbb{R}^r$ such that $\nabla_x\mathcal{L}(x_*,\lambda_*)$ is the null vector. So, in possession of this vector λ_*, we define $g:\mathbb{R}^n\to\mathbb{R}^{n-r}$, where

$$g(x):=Z_x^{\triangleleft T}\nabla_x\mathcal{L}(x,\lambda_*)=Z_x^{\triangleleft T}\nabla\phi_{r+1}(x). \tag{4.16}$$

Moreover, for not overloading the proofs that will follow, we also define

$$A_k:=I_n-Z_{x_k}^{\triangleleft}\hat{H}_k^{-1}Z_{x_k}^{\triangleleft\,T}H_k, \tag{4.17}$$

with

$$\hat{H}_k:=Z_{x_k}^{\triangleleft\,T}H_kZ_{x_k}^{\triangleleft}.$$

Below, we present a theorem that establishes the exact solution $d_{k,\bar{l}_k}$ obtained in (4.4) when it is equivalent to (4.12). For this result and the subsequent ones, we define

$$\tau_{k,\bar{l}_k}:=\max_{1\leqslant i\leqslant r+1}\left\|x_{k,i}^{\bar{l}_k}-x_k\right\|.$$

Theorem 4.2. *Suppose we are at a fixed iteration k of GraFuS and at the last inner iteration indexed by $\bar{l}_k$. Then, if the hypotheses H1, H2 and H3 hold, and $x_k\in\mathcal{W}$, we have that*

$$d_{k,\bar{l}_k}=d_{k,\bar{l}_k}^U+d_{k,\bar{l}_k}^V,$$

where

$$d_{k,\bar{l}_k}^U:=-Z_{x_k}^{\triangleleft}\hat{H}_k^{-1}g(x_k)+\rho_k^U\quad\text{and}\quad d_{k,\bar{l}_k}^V:=-A_kJ_{x_k}^{\triangleleft}\Phi(x_k)+\rho_k^V,$$

with

$$\rho_k^U=-Z_{x_k}^{\triangleleft}\hat{H}_k^{-1}Z_{x_k}^{\triangleleft\,T}\overline{\rho}_k\quad\text{and}\quad\rho_k^V=-A_kJ_{x_k}^{\triangleleft}\hat{\rho}_k,$$

for some $\overline{\rho}_k\in\mathbb{R}^n$ and $\hat{\rho}_k\in\mathbb{R}^r$ satisfying

$$\|\overline{\rho}_k\|=O\left(\tau_{k,\bar{l}_k}\right)\quad\text{and}\quad\|\hat{\rho}_k\|=O\left(\tau_{k,\bar{l}_k}^2\right)+O\left(\tau_{k,\bar{l}_k}\right)O\left(\nu_k\right).$$

Proof. First, we consider the Karush-Kuhn-Tucker conditions of problem (4.12), which tell us that the solution $d_{k,\bar{l}_k}$ must satisfy

$$\tilde{\Phi}_k+\tilde{J}_kd_{k,\bar{l}_k}=0 \tag{4.18}$$

and

$$\nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)+H_kd_{k,\bar{l}_k}+\tilde{J}_k^T\tilde{\lambda}=0, \tag{4.19}$$

for some $\tilde{\lambda} \in \mathbb{R}^r$. Since the functions that comprise f satisfy $\phi_i \in C^2$, for $i \in \{1, \ldots, p\}$, we have, by relations (4.18) and (4.19) and by

$$\|d_{k,\bar{l}_k}\|_\infty \leqslant \Delta_{k,\bar{l}_k} \leqslant \gamma_\Delta \nu_k,$$

that

$$\begin{aligned} \Phi(x_k) + J_{x_k} d_{k,\bar{l}_k} + \left[\tilde{\Phi}_k - \Phi(x_k)\right] + \left[\tilde{J}_k - J_{x_k}\right] d_{k,\bar{l}_k} = 0 \\ \Phi(x_k) + J_{x_k} d_{k,\bar{l}_k} + \hat{\rho}_k = 0 \end{aligned} \tag{4.20}$$

and

$$\nabla \phi_{r+1}(x_k) + H_k d_{k,\bar{l}_k} + J_{x_k}^T \tilde{\lambda} + \overline{\rho}_k = 0, \tag{4.21}$$

where $\|\hat{\rho}_k\| = O\left(\tau_{k,\bar{l}_k}^2\right) + O\left(\tau_{k,\bar{l}_k}\right) O(\nu_k)$ and $\|\overline{\rho}_k\| = O\left(\tau_{k,\bar{l}_k}\right)$. Then, because $A_k J_{x_k}^{\lhd}$ is a right inverse for J_{x_k} (see [2, Section 14.2]), it is possible to decompose $\mathbb{R}^n$ in two subspaces generated by the columns of $Z_{x_k}^{\lhd}$ and $A_k J_{x_k}^{\lhd}$. As a consequence, we can consider two vectors $d_{k,\bar{l}_k}^U$ and $d_{k,\bar{l}_k}^V$ such that there exist α_U and α_V that imply

$$d_{k,\bar{l}_k} = d_{k,\bar{l}_k}^U + d_{k,\bar{l}_k}^V,$$

with

$$d_{k,\bar{l}_k}^U = Z_{x_k}^{\lhd} \alpha_U \quad \text{and} \quad d_{k,\bar{l}_k}^V = A_k J_{x_k}^{\lhd} \alpha_V.$$

Hence, looking at relation (4.20), we obtain that

$$\alpha_V = -\Phi(x_k) - \hat{\rho}_k,$$

which yields

$$d_{k,\bar{l}_k}^V = -A_k J_{x_k}^{\lhd} \Phi(x_k) + \rho_k^V, \quad \text{with} \quad \rho_k^V = -A_k J_{x_k}^{\lhd} \hat{\rho}_k.$$

Finally, pre-multiplying the relation (4.21) by ${Z_{x_k}^{\lhd}}^T$, we have

$$g(x_k) + {Z_{x_k}^{\lhd}}^T H_k \left[Z_{x_k}^{\lhd} \alpha_U - A_k J_{x_k}^{\lhd} \left(\Phi(x_k) + \hat{\rho}_k\right)\right] + {Z_{x_k}^{\lhd}}^T \overline{\rho}_k = 0.$$

Then, since ${Z_{x_k}^{\lhd}}^T H_k A_k = 0$, we complete the proof by noticing that

$$\alpha_U = -\hat{H}_k^{-1} g(x_k) - \hat{H}_k^{-1} {Z_{x_k}^{\lhd}}^T \overline{\rho}_k \Rightarrow d_{k,\bar{l}_k}^U = -Z_{x_k}^{\lhd} \hat{H}_k^{-1} g(x_k) + \rho_k^U,$$

where $\rho_k^U = -Z_{x_k}^{\lhd} \hat{H}_k^{-1} {Z_{x_k}^{\lhd}}^T \overline{\rho}_k$. □

With this theorem in hand, we are able to prove a simple corollary.

Corollary 4.1. *Under the assumptions of Theorem 4.2, we have that*

$$\|\Phi(x_{k+1})\| = O(\nu_k^2).$$

Proof. Since $\phi_i \in C^2$, $\|d_{k,\bar{l}_k}\| = O(\nu_k)$ and $\tau_{k,\bar{l}_k} = O(\nu_k)$, it yields that

$$\begin{aligned}\|\Phi(x_{k+1})\| &\leqslant \|\Phi(x_k) + J_{x_k} d_{k,\bar{l}_k}\| + O(\nu_k^2)\\ &\leqslant \|\Phi(x_k) - \Phi(x_k)\| + \|\hat{\rho}_k\| + O(\nu_k^2)\\ &= O(\nu_k^2),\end{aligned}$$

which is the desired result. □

The previous statement leaves us with an important observation: when GraFuS samples under hypothesis H1, H2 and H3, the homogeneous system $\Phi(x) = 0$ is quickly satisfied, since ν_k is associated with our optimality certificate (notice that the term $O(\nu_k^2)$ in Corollary 4.1 could also be changed to $o(\epsilon_{k,\bar{l}_k})$ or $o(\Delta_{k,\bar{l}_k})$ without losing validity).

Finally, we are able to prove the most important result of this subsection.

Theorem 4.3. *Suppose that $x_k \to x_*$, where $x_* \in \mathbb{R}^n$ is a local minimizer for f presented in (4.1). Assume that, for iterations with indices in an infinite set $\mathcal{K} \subset \mathbb{N}$, hypotheses H1, H2 and H3 hold and $x_k \in \mathcal{W}$. Also, suppose that the maps*

$$Z^{\triangleleft}: \begin{array}{ccc}\mathbb{R}^n & \longrightarrow & \mathbb{R}^{n\times(n-r)}\\ x & \longmapsto & Z_x^{\triangleleft}\end{array} \quad \text{and} \quad J^{\triangleleft}: \begin{array}{ccc}\mathbb{R}^n & \longrightarrow & \mathbb{R}^{n\times r}\\ x & \longmapsto & J_x^{\triangleleft}\end{array}$$

are all Lipschitz continuous functions close to x_* and that the reduced gradient given in (4.16) satisfies $g \in C^1$ with g' being also a Lipschitz continuous function close to x_*. Moreover, assume that $H_k \to H_*$ is such that

$$H_* = \nabla^2_{xx}\mathcal{L}(x_*, \lambda_*) + \gamma J_{x_*}^T J_{x_*}, \quad \text{for some } \gamma \geqslant 0. \tag{4.22}$$

Additionally, suppose that close to x_* we have that $\|H_k - H_*\| = O(\|x_k - x_*\|)$. Then, the following relation holds

$$\|x_{k+1} - x_*\| = O(\|x_k - x_*\|^2) + \rho_k^U + \rho_k^V, \quad \text{for } k \in \mathcal{K}.$$

Proof. First, let us define $\tilde{x}_{k+1} := x_k + d^V_{k,\bar{l}_k}$, with $k \in \mathcal{K}$. Now, observe that, from the definition (4.17), for x_k close enough to x_*, we have $\|A_k - A_*\| = O(\|x_k - x_*\|)$, where

$$A_* := I_n - Z^{\triangleleft}_{x_*}\hat{H}_*^{-1} {Z^{\triangleleft}_{x_*}}^T H_*, \quad \text{with } \hat{H}_* := {Z^{\triangleleft}_{x_*}}^T H_* Z^{\triangleleft}_{x_*}.$$

Using this fact, considering the Taylor expansion of the map Φ around x_* in the relation $(*)$ below and noticing that $J^{\triangleleft}$ is Lipschitz continuous and a bounded map around x_* in $(**)$, we have for a sufficiently small neighborhood of x_* that

$$\begin{aligned}\tilde{x}_{k+1} - x_* &= x_k - x_* - A_k J^{\triangleleft}_{x_k}\Phi(x_k) + \rho_k^V\\ &\overset{(*)}{=} x_k - x_* - A_k J^{\triangleleft}_{x_k} J_{x_*}(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho_k^V\\ &= x_k - x_* - A_* J^{\triangleleft}_{x_*} J_{x_*}(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho_k^V\\ &\quad - \left[A_k\left(J^{\triangleleft}_{x_k} - J^{\triangleleft}_{x_*}\right) + (A_k - A_*)\, J^{\triangleleft}_{x_*}\right] J_{x_*}(x_k - x_*)\\ &\overset{(**)}{=} x_k - x_* - A_* J^{\triangleleft}_{x_*} J_{x_*}(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho_k^V.\end{aligned}$$

Consequently, taking into account the relation (see [2, Section 14.5])

$$g'(x_*) = {Z_{x_*}^{\triangleleft}}^T \nabla^2_{xx}\mathcal{L}(x_*, \lambda_*)$$

in ($\bullet$), the Lipschitz property around x_* of the maps $Z^{\triangleleft}$ and $\hat{H}^{-1}$ in ($\bullet\bullet$), the relation (4.22) in ($\blacktriangle$) and the relation (4.14) in ($\blacktriangle\blacktriangle$), we have

$$\begin{aligned}
x_{k+1} - x_* &= \tilde{x}_{k+1} - x_* - Z_{x_k}^{\triangleleft}\hat{H}_k^{-1} g(x_k) + \rho_k^U \\
&\overset{(\bullet)}{=} \tilde{x}_{k+1} - x_* - Z_{x_k}^{\triangleleft}\hat{H}_k^{-1}{Z_{x_*}^{\triangleleft}}^T \nabla^2_{xx}\mathcal{L}(x_*, \lambda_*)(x_k - x_*) \\
&\quad + O(\|x_k - x_*\|^2) + \rho_k^U \\
&\overset{(\bullet\bullet)}{=} \tilde{x}_{k+1} - x_* - Z_{x_*}^{\triangleleft}\hat{H}_*^{-1}{Z_{x_*}^{\triangleleft}}^T \nabla^2_{xx}\mathcal{L}(x_*, \lambda_*)(x_k - x_*) \\
&\quad + O(\|x_k - x_*\|^2) + \rho_k^U \\
&\overset{(\blacktriangle)}{=} \tilde{x}_{k+1} - x_* - Z_{x_*}^{\triangleleft}\hat{H}_*^{-1}{Z_{x_*}^{\triangleleft}}^T H_*(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho_k^U \\
&= A_*(x_k - x_*) - A_* J_{x_*}^{\triangleleft} J_{x_*}(x_k - x_*) + O(\|x_k - x_*\|^2) \\
&\quad + \rho_k^U + \rho_k^V \\
&= A_*(I - J_{x_*}^{\triangleleft} J_{x_*})(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho_k^U + \rho_k^V \\
&\overset{(\blacktriangle\blacktriangle)}{=} A_* Z_{x_*}^{\triangleleft} Z_{x_*}(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho_k^U + \rho_k^V .
\end{aligned}$$

Hence, since $A_* Z_{x_*}^{\triangleleft} = 0$, it yields that

$$\|x_{k+1} - x_*\| = O(\|x_k - x_*\|^2) + \rho_k^U + \rho_k^V ,$$

which concludes the proof. □

With the above result, we see that the only term that might prevent the algorithm to move superlinearly to the solution x_* is $\tau_{k,\bar{l}_k}$. Therefore, since $\tau_{k,\bar{l}_k}$ is intimately linked to the sampling radius, it would be interesting to have the following relation: $\epsilon_{k,\bar{l}_k}^{\sigma_{\bar{l}_k}} = o(\|x_k - x_*\|)$. If this last equation holds, the algorithm moves superlinearly to the solution as $k \in \mathcal{K}$. It is clear that imposing that relation to $\epsilon_{k,\bar{l}_k}^{\sigma_{\bar{l}_k}}$ is impossible, since this demands to know x_*. However, taking a careful look at the Karush-Kuhn-Tucker conditions of (4.4), we have that

$$d_{k,\bar{l}_k} = -H_k^{-1}\left(G_{k,\bar{l}_k}\lambda_{k,\bar{l}_k} + \omega_{k,\bar{l}_k}\right). \tag{4.23}$$

Therefore, considering $\epsilon_{k,\bar{l}_k} = O(\nu_k)$ and by the way ν_k is defined in GraFuS, we see (specially when $\omega_{k-1,\bar{l}_{k-1}} = 0$, i.e., when $d_{k-1,\bar{l}_{k-1}} < \Delta_{k-1,\bar{l}_{k-1}}$) that $\epsilon_{k,\bar{l}_k}$ is a reasonable approximation of $\|d_{k-1,\bar{l}_{k-1}}\|$. On the other hand, $\|d_{k-1,\bar{l}_{k-1}}\|$ can be seen as a measure of how far the algorithm is from $\|x_k - x_*\|$ (considering $x_k \to x_*$). So, since the sampled points are chosen in $\mathcal{B}(x_k, \gamma_\epsilon \epsilon_{k,\bar{l}_k}^{\sigma_{\bar{l}_k}})$, it is not absurd to expect that, for an appropriate value of $\sigma_{\bar{l}_k} \in [1, 2]$, the following will hold for a reasonable amount of times

$$\tau_{k,\bar{l}_k} = o(\|x_k - x_*\|). \tag{4.24}$$

One can argue that it would be better to let σ_l be greater than two in order to have a smaller sampling radius, and consequently, to increase the chance of (4.24) to happen. However, we cannot forget that to allow the possibility of moving superlinearly to the solution, the method must have a good set of sampled points. A tiny sampling radius might give us a bad representation of the function by e.g. sampling points where just one ϕ_i assumes the maximum value. Therefore, a trade-off must be assumed between these two conflicting needs.

Remark 4.3. *We stress that* $\epsilon_{k,\bar{l}_k}^{\sigma_{\bar{l}_k}} = o(\|x_k - x_*\|)$ *is a desirable result, but by no means, it is a necessary condition for a rapid movement towards the solution of the optimization problem. Let us consider that* $\epsilon_{k,\bar{l}_k}^{\sigma_{\bar{l}_k}}$ *is large. Even for this case, since we have a uniform sample around* x_k*, a sampling over the set* $\mathcal{B}(x_k, o(\|x_k - x_*\|))$ *is an event that occurs with probability greater than zero, which yields that a superlinear movement is a real possibility (considering a good set of sampled points, which shows the importance of Corollary 4.1). As a result, a good approximation of the value* $\|x_k - x_*\|$ *just allows the method to increase the probability of* (4.24) *to happen. Finally, we also highlight that for the case that* $V = \mathbb{R}^n$*, the equality* (4.24) *can be replaced by* $\tau_{k,\bar{l}_k} = O(\|x_k - x_*\|)$ *without affecting the superlinear convergence (to see this, just notice the properties of* ρ_k^V*).*

Remark 4.4. *All the local convergence results were made assuming* $r \in \{1, \ldots, n-1\}$*. For the case* $r = 0$*, we have that the method is approaching a point at which the function* f *is smooth in the whole space. For such a situation, it is straightforward to see that the direction* $d_{k,\bar{l}_k}$ *will have only the* U *component, i.e.,* $d_{k,\bar{l}_k} = d_{k,\bar{l}_k}^U$ *with* $Z_x^{\triangleleft} = I_n$ *for all* x *around* x_**. Now, considering* $r = n$*, we see that the method is approaching a point at which* f *is nonsmooth in any direction. For that case, it is also clear to see that the direction* $d_{k,\bar{l}_k}$ *will have only the* V *component, i.e.,* $d_{k,\bar{l}_k} = d_{k,\bar{l}_k}^V$ *with* $A_k \equiv I_n$ *for all* x_k *around* x_**. Therefore, in these two cases, the method will also move superlinearly if the sampling radius is assumed to be small enough and if the algorithm has a good set of sampled points (for* $r = n$*).*

4.3 Numerical results

Since a superlinear move is dependent on a good set of sampled points, one might think that the necessary hypotheses will be true just a few times during the execution of the method. This subsection has the intent to show that a rapid move to the solution is frequent enough to speed up the local convergence. However, by no means we have the ambition to present an extensive set of tests nor to recommend our method over any other one. Here, our main goal is to have numerical results that present to the reader a proof-of-concept. Finally, we also aim at showing that one can expect global convergence for more general problems than the ones considered in our theoretical results.

For all functions we have chosen random starting points such that $\|x_0\|_\infty \leqslant 2$ and solved each of them twenty times in order to have statistical relevance of the results. The comparable figures were plotted using the median and quartiles (25% and 75%) of those twenty runs and also the best function value f_* obtained by both methods in all of the runs.

We have solved each optimization problem with two algorithms: (i) the GS method presented by the original authors [6] but with a nonnormalized search direction (a variant introduced by Kiwiel [27], that has the advantage to asymptotically recover the steepest descent method when applied to smooth functions) and (ii) the GraFuS method. We have used the original GS implementation without any modification (with the exception of using a nonnormalized search direction). The parameter values used in Algorithm 2 were: $m = 2n$; $\nu_0 = 10^{-3}$; $\nu_{\text{opt}} = 10^{-6}$; $\gamma_\epsilon = 4\sqrt{n}$; $\gamma_\Delta = 5$; $\iota = 2$; $\rho = 10^{-8}$ and $\theta = 0.5$. The value of σ_l in Step 1 was set as $\sigma_l = 1 + 0.5^{(l+2)/4} \mod (l+1, 2)$.

Along the analysis presented in this chapter, we suppose that the sampling procedure is uniform. However, we must remind the reader that when the number of variables, n, grows, an uniform distribution will prioritize the points near the frontier of the Euclidean ball. This happens because the volume of an n-dimensional ball is concentrated almost entirely around the frontier, when n is sufficiently large. This property has a direct implication on our method, since the chance to satisfy (4.24) with a large sampling radius can be dramatically reduced. Hence, for the numerical results presented here, we have first uniformly sampled an unitary vector and then, we have resized the length of this vector by an uniform sample in the interval $[0, \gamma_\epsilon \epsilon_{k,l}^\sigma]$. Such a procedure does not invalidate the proofs presented previously, i.e., the global convergence result is not affected.

An important point that must be stressed here is that the iterations of GraFuS are more expensive than those of GS. While the GS routine finds a search direction and does an Armijo line search to find the next iterate, GraFuS constantly solves quadratic programming problems until it finds a good set of sampled points and a good trust region to move. Therefore, one could take advantage of the way GS was designed as a threshold to start performing GraFuS iterations, deciding if the current iterate is close to the solution indirectly by means of the size of the current sampling radius. As a result, we only start to run the GraFuS algorithm after the second reduction of the sampling radius in GS (i.e. when $\epsilon_k < 10^{-2}$), and that is the reason why in the figures that follow below, we see that in the first iterations both methods remain together.

Finally, the way we have chosen the matrices H_k is a delicate matter and, for that reason, we have reserved the following subsection to explain our procedure. It is worth pointing out that we have used BFGS ideas to update the matrices, but we do not have any theoretical guarantee that the matrices H_k will converge to a matrix of the form presented in (4.22). Nevertheless, the choice on how we update the matrices has a strong

foundation, since it uses the same reasoning of a Sequential Quadratic Programming (SQP) updating [16] for the optimization problem that appears in (4.13).

4.3.1 H_k updates in GraFuS method

As we have seen in the last section, if some hypotheses are satisfied, it is possible to see the quadratic programming problem that is solved in every iteration of GraFuS as a smooth constrained optimization problem. Moreover, the matrix that we would like to approximate (at least in its null space) is the Hessian of (4.15). Therefore, a natural attempt to reach that goal is to update the positive definite matrix H_k as it is done in SQP routines. In other words, it would be desirable to have the following relation

$$H_k(x_+ - x_-) = \nabla_x \mathcal{L}(x_+, \lambda_+) - \nabla_x \mathcal{L}(x_-, \lambda_-),$$

where $\mathcal{L}$ is the Lagrangian function defined in (4.15) and λ_+ and λ_- are vectors that try to approximate the multiplier λ_* that fulfills (4.16). In addition,

$$\begin{aligned}\nabla_x \mathcal{L}(x, \lambda) &= \nabla \phi_{r+1}(x) + \sum_{i=1}^{r} \lambda_i (\nabla \phi_i(x) - \nabla \phi_{r+1}(x)) \\ &= \left(1 - \sum_{i=1}^{r} \lambda_i\right) \nabla \phi_{r+1}(x) + \sum_{i=1}^{r} \lambda_i \nabla \phi_i(x).\end{aligned}$$

Therefore, defining $\hat{\lambda} \in \mathbb{R}^{r+1}$ as $\hat{\lambda}_i = \lambda_i$, for $i \in \{1, \ldots, r\}$, and

$$\hat{\lambda}_{r+1} = 1 - \sum_{i=1}^{r} \lambda_i,$$

we have $e^T \hat{\lambda} = 1$ and one can rewrite $\nabla_x \mathcal{L}(x, \lambda) = \hat{G}\hat{\lambda}$, where

$$\hat{G} := [\nabla \phi_1(x) \ \ldots \nabla \phi_{r+1}(x)].$$

Hence, if in two fixed outer and inner iterations k_+, k_- and l_+, l_-, respectively, we have that hypotheses H1, H2 and H3 are satisfied, it is natural to ask that the following secant relationship holds

$$H_k(x_{k_+} - x_{k_-}) = G_{k_+,l_+} \lambda_{k_+,l_+} - G_{k_-,l_-} \lambda_{k_-,l_-}.$$

The problem here is how one can identify if the aforementioned hypotheses hold. In fact, although there is no straightforward response, we know that a good set of sampled points is associated with a small norm of the convex combination of its gradients. Hence, a good strategy would be to update the matrix H_k only if such a condition is verified.

Based on the previous reasoning, we present next the routine that provides the sequence of matrices H_k that are used within GraFuS.

Step 0. Start setting $H = I$ and let the GraFuS algorithm run until it finds two outer iterations k_+, k_- such that

$$\left\|G_{k_+,\bar{l}_{k_+}}\lambda_{k_+,\bar{l}_{k_+}}\right\|_\infty \leqslant 10\nu_{k_+} \quad \text{and} \quad \left\|G_{k_-,\bar{l}_{k_-}}\lambda_{k_-,\bar{l}_{k_-}}\right\|_\infty \leqslant 10\nu_{k_-}.$$

Set

$$x_+ := x_{k_+} \text{ and } x_- := x_{k_-};$$

$$v_+ := G_{k_+,\bar{l}_{k_+}}\lambda_{k_+,\bar{l}_{k_+}} \text{ and } v_- := G_{k_-,\bar{l}_{k_-}}\lambda_{k_-,\bar{l}_{k_-}}.$$

Step 1. Set $p := x_+ - x_-$ and $q := v_+ - v_-$. If $q^Tp < 0.2p^THp$ then compute a new vector q by Powell's correction (see [2, Subsection 18.2]).

Step 2. Update H:

$$H \leftarrow H - \frac{Hpp^TH}{p^THp} + \frac{qq^T}{q^Tp}.$$

Step 3. Use the subsequent matrices H_k as H until the GraFuS algorithm finds another iteration $\hat{k}$ such that

$$\left\|G_{\hat{k},\bar{l}_{\hat{k}}}\lambda_{\hat{k},\bar{l}_{\hat{k}}}\right\|_\infty \leqslant 10\nu_{\hat{k}}.$$

Then, $x_- \leftarrow x_+$, $x_+ \leftarrow x_{\hat{k}}$, $v_- \leftarrow v_+$, $v_+ \leftarrow G_{\hat{k},\bar{l}_{\hat{k}}}\lambda_{\hat{k},\bar{l}_{\hat{k}}}$. Go back to Step 1.

Clearly, other ways of updating H_k are possible. Indeed, even the pure BFGS update as considered in [32] can be performed (although, in such a case, we have to assume that for all iterates the function f will be differentiable and Assumption 2.1 will no longer be satisfied). For us, this previous routine was the one that seemed more reasonable in light of assumptions H1, H2 and H3, and generated good numerical results.

Below, we present the functions that were solved and divide them in different categories. The black line plot in the following figures represents the GS method, whereas the grey continuous one with $\diamondsuit$ marks is the GraFuS method. In addition, we must stress that although the optimality certificates of Algorithms 1 and 2 are very similar, they are not the same (specially because the quadratic programming problem of each method is different). Therefore, one might be more rigorous than the other one. Thus, although in most problems the GraFuS method appears to be closer to the solution, this does not mean that GS is not able to reach the same precision (maybe a tighter optimality parameter would allow it).

Additionally, as a tool for assessing how fast our method goes towards the solution, we have represented the ratio

$$\frac{f(x_{k+1}) - f_*}{f(x_k) - f_*}$$

with color scales along the plotted curves of GraFuS, where the red hue stands for a ratio close to zero and the blue color for the values near one.

4.3.2 Test functions with $V = \mathbb{R}^n$

We present two nonconvex and nonsmooth functions [18] that, at the solution point, have the whole space $\mathbb{R}^n$ as the V space:

F1) Active faces (defined for all number of variables n)

$$f(x) = \max\left\{ g\left(-\sum_{i=1}^{n} x_i \right), \max_{1 \leqslant i \leqslant n} \{g(x_i)\} \right\}, \quad \text{with} \quad g(z) = \log(|z| + 1);$$

F2) Chained Mifflin 2 (defined for all number of variables $n \geqslant 2$)

$$f(x) = \sum_{i=1}^{n-1} \left(-x_i + 2(x_i^2 + x_{i+1}^2 - 1) + 1.75|x_i^2 + x_{i+1}^2 - 1| \right).$$

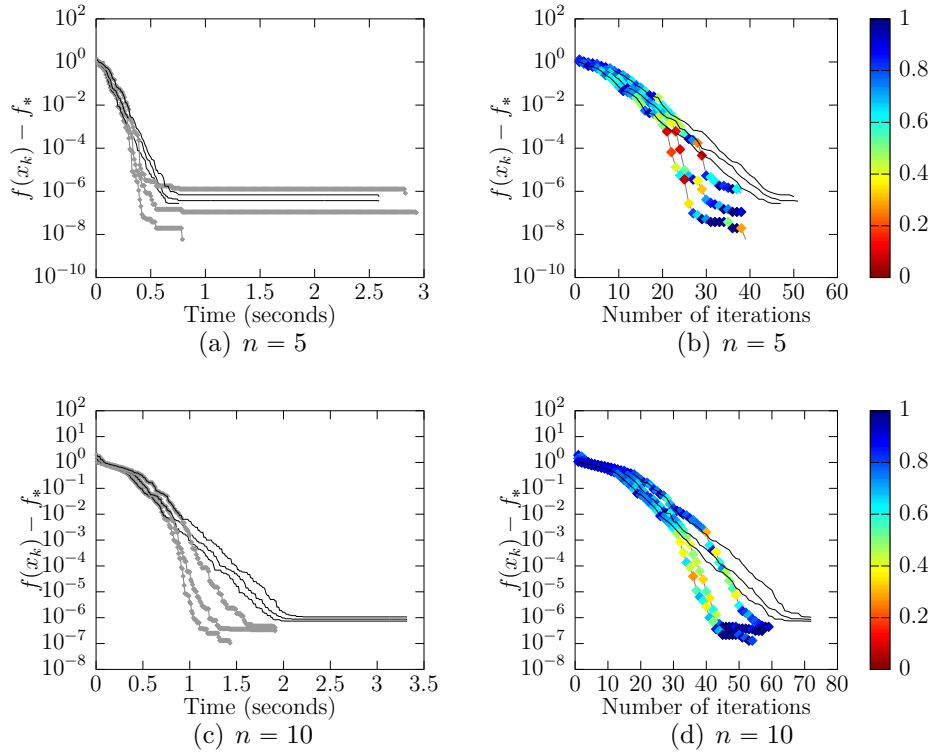


Figure 16 – Medians and quartiles of twenty runs of GS and GraFuS methods for function **F1**. For both number of variables we have $x_* = 0$.

Before we proceed, an important observation must be taken into consideration. Suppose that one has $f(x) = \max\{g_1(x), g_2(x)\} + \max\{h_1(x), h_2(x)\}$. Therefore, it is possible to turn the previous function into a maximum of functions by just noticing that f can be written as

$$f(x) = \max\{g_1(x) + h_1(x), g_1(x) + h_2(x), g_2(x) + h_1(x), g_2(x) + h_2(x)\}.$$

In other words, f is the maximum of all possible combinations of g_1 and g_2 with h_1 and h_2. With the generalization of this reasoning and remembering that $|x| = \max\{-x, x\}$, we

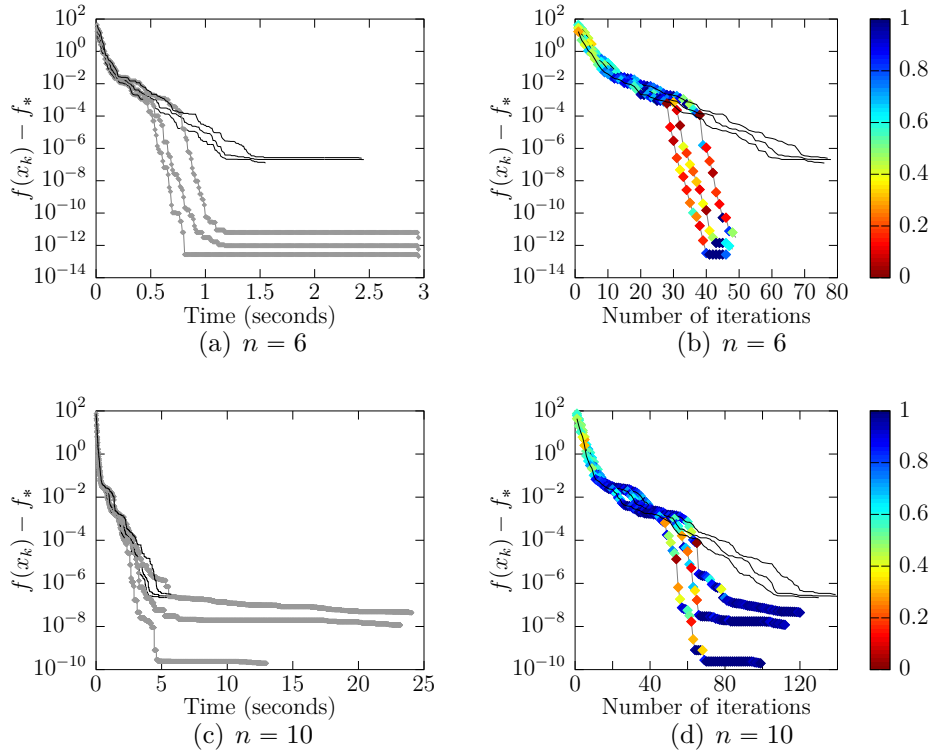


Figure 17 – Medians and quartiles of twenty runs of GS and GraFuS methods for function **F2**. For $n = 6$ and $n = 10$ we have, respectively, $x_* \approx (0.8152, 0.5792, 0.7747, 0.6323, 0.7747, 0.0000)^T$ and $x_* \approx (0.8152, 0.5792, 0.7362, 0.6767, 0.7362, 0.6767, 0.7362, 0.6767, 0.7362, 0)^T$.

see, at least in a close neighborhood of x_*, that **F1** and **F2** can be viewed as maximum of smooth functions.

A closer look at the expressions of those functions reveals to us that the number of active functions at their solutions have more than $n + 1$ active functions. Therefore, Assumption 4.1 does not hold for the functions **F1** and **F2**. Fortunately, this fact does not prevent GraFuS to converge for both functions (see Figures 16 and 17).

The good behavior in the absence of the validity of Assumption 4.1 was somehow expected. In fact, if one can guarantee that without this assumption we still have open sets where each active function assumes the maximum, the probability that the sampled points be in regions of the domain where just some specific combination of $n + 1$ functions reaches the maximum is strictly positive, and consequently, the results hold.

Finally, looking at the plots that compare iterations versus the distance of the current function value to f_*, in general, we can observe some rapid moves to the solution as expected, with the exception of Figure 16 (d), where a rapid movement towards the solution is not detected. However, it is possible to adjust the parameters of GraFuS in order to have a better behavior of our method for this instance. When one looks to convergence over time, it is possible to see that GraFuS is competitive with the well established GS

method.

Remark 4.5. *The functions inside the subsection of objective functions with multiple stationary points do also satisfy $V = \mathbb{R}^n$. However, we have chosen to separate them from **F1** and **F2** because they have an additional property.*

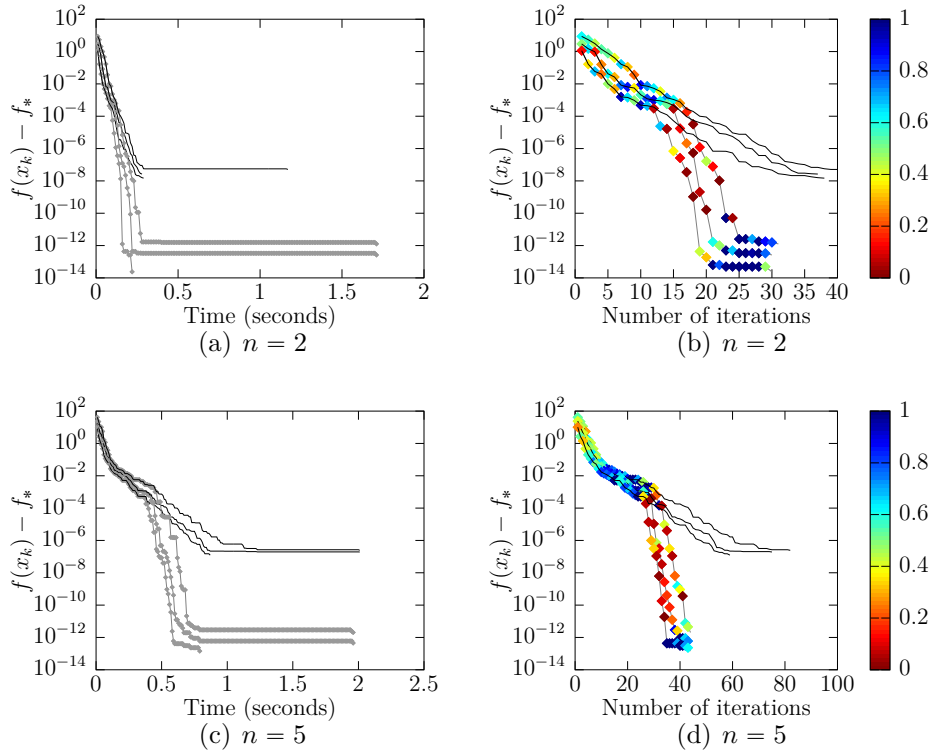


Figure 18 – Medians and quartiles of twenty runs of GS and GraFuS methods for function **F2**. For $n = 2$ and $n = 5$ we have, respectively, $x_* = (1, 0)^T$ and $x_* \approx (0.8152, 0.5792, 0.7071, 0.7071, 0)^T$.

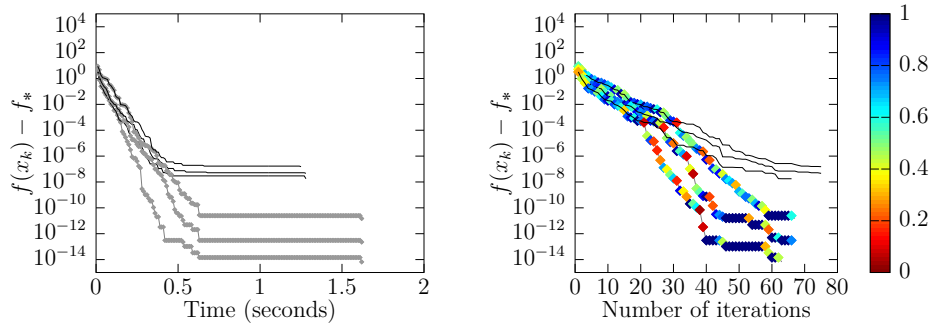


Figure 19 – Medians and quartiles of twenty runs of GS and GraFuS methods for function **F3** (with $n = 2$). We have $x_* = (1, 0.5)^T$.

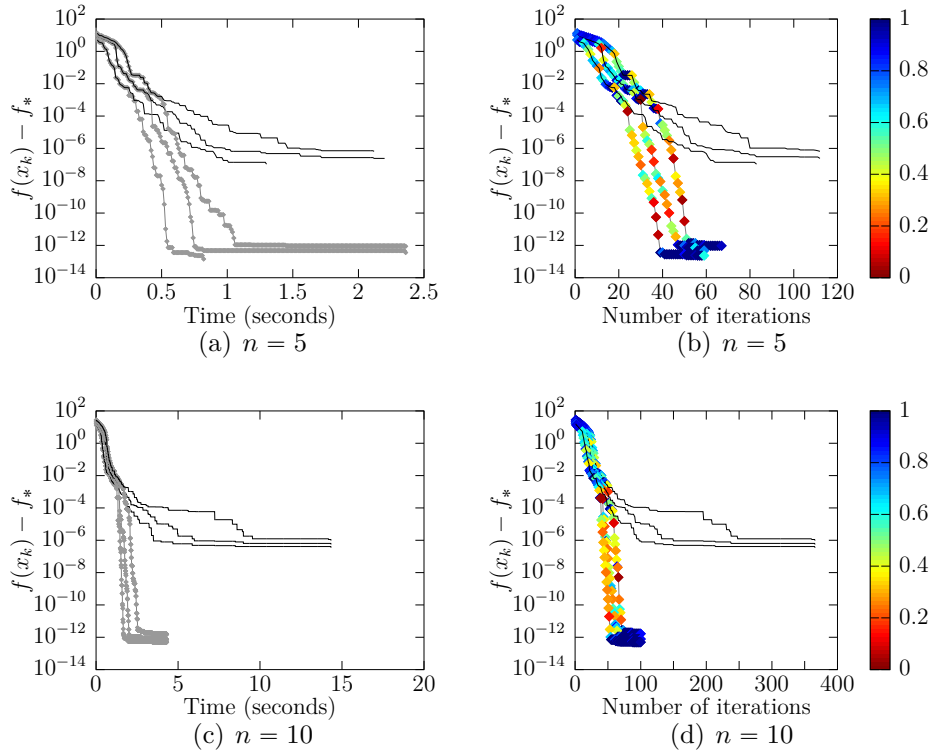


Figure 20 – Medians and quartiles of twenty runs of GS and GraFuS methods for function **F4**. For both number of variables we have $x_* = 0$.

4.3.3 Test functions with $V \neq \mathbb{R}^n$

In the previous subsection we only presented functions for which the $\dim\{U\} = 0$ at the stationary points. In opposite direction, here we show and solve functions that, at the stationary points, can behave in a smooth way for some directions. We have considered the following functions [18, 47]: **F2** presented previously (but now with $n \in \{2, 5\}$) and

F3) Generalized Rosenbrock function (defined for all number of variables $n \geqslant 2$)

$$f(x) = \sum_{i=1}^{n-1} \left(\frac{10i}{n} \left| x_{i+1} - \frac{i}{n} x_i^2 \right| + \frac{i}{n} (1 - x_i)^2 \right).$$

F4) Chained crescent I (defined for all number of variables $n \geqslant 2$)

$$f(x) = \max \left\{ \sum_{i=1}^{n-1} \left(x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1 \right), \right.$$
$$\left. \sum_{i=1}^{n-1} \left(-x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1 \right) \right\}.$$

It is worth pointing out that for **F2** and **F3**, we have set, respectively, $n \in \{2, 5\}$ and $n = 2$ only. This was done in order to maintain a dimension greater than zero for the U space at the solution point. As a counterpart, there is no restriction on the dimension

of **F4**, and therefore, we have solved instances with $n = 5$ and $n = 10$. The results can be seen in Figures 18 - 20 and the rapid convergence behavior is also observed in some iterations of GraFuS.

4.3.4 Test functions with multiple stationary points

In order to have a broader illustrative class of functions, we minimize in this subsection two nonconvex and nonsmooth functions with multiple stationary points [18,47]:

F5) Chained crescent II (defined for all number of variables $n \geqslant 2$)

$$f(x) = \sum_{i=1}^{n-1} \max\left\{\left(x_i^2 + (x_{i+1}-1)^2 + x_{i+1} - 1\right),\right.$$
$$\left.\left(-x_i^2 - (x_{i+1}-1)^2 + x_{i+1} + 1\right)\right\};$$

F6) Problem 17 of Test 29 of [47] (defined for all n multiple of 5)

$$f(x) = \max_{1\leqslant i\leqslant n}\left\{\left|5 - (j+1)(1-\cos x_i) - \operatorname{sen} x_i - \sum_{k=5j+1}^{5j+5}\cos x_k\right|\right\},$$

with $j = \lfloor (i-1/5)\rfloor$.

The results can be found in Figures 21 and 22. Again, it is possible to find iterates for which the algorithm moves fast to the solution, enlarging the results previously obtained.

4.3.5 Test functions without an appropriate maximum representation

The next functions can be seen in [18,47] and they cannot be written as the maximum of sufficiently smooth functions:

F7) Nonsmooth generalization of Brown function 2 (defined for all number of variables $n \geqslant 2$)

$$f(x) = \sum_{i=1}^{n-1}\left(|x_i|^{x_{i+1}^2+1} + |x_{i+1}|^{x_i^2+1}\right);$$

F8) Nonsmooth and nonconvex toy problem (defined for all number of variables $n \geqslant 2$)

$$f(x) = \sqrt{g(x)}, \quad \text{with} \quad g(x) = \delta + \sqrt{x^T A x} + x^T B x,$$

where $\delta \in (0,1)$ is a fixed parameter, $A = \operatorname{diag}(1,0,1,0,\ldots)$ and $B = \operatorname{diag}(1,\ldots,1/n^2)$.

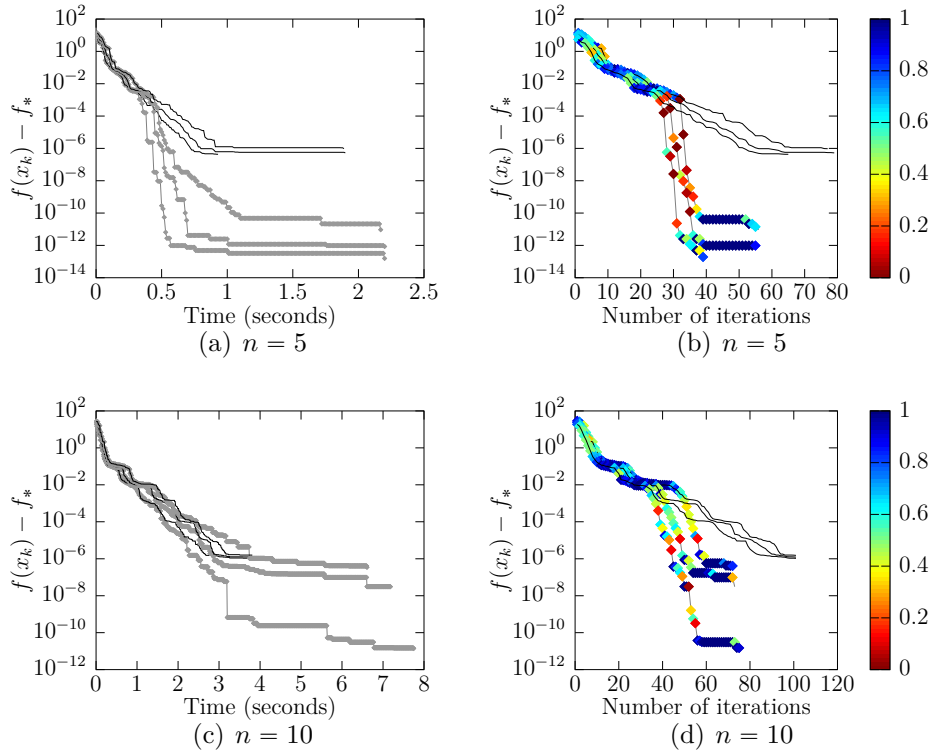


Figure 21 – Medians and quartiles of twenty runs of GS and GraFuS methods for function **F5**. For both number of variables we have $x_* = 0$.

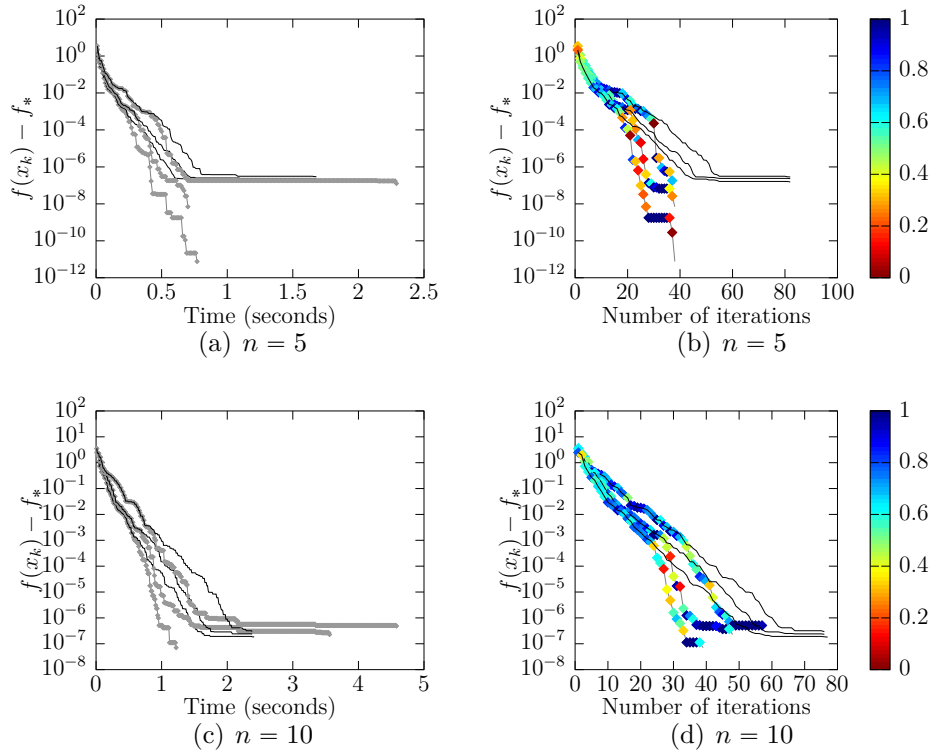


Figure 22 – Medians and quartiles of twenty runs of GS and GraFuS methods for function **F6**. For both number of variables we have $x_* = 0$.

For function **F7**, one may argue that it is not possible to have a maximum representation with functions of class C^1. Indeed, let us consider the function $h(a,b) = a^{(1+b^2)}$, for $a \geqslant 0$. Then, it yields that

$$\lim_{\varepsilon \downarrow 0} \frac{\partial h}{\partial a}(\varepsilon, \varepsilon) = \lim_{\varepsilon \downarrow 0} (1+\varepsilon^2)\varepsilon^{\varepsilon^2} = 1;$$

$$\lim_{\varepsilon \downarrow 0} \frac{\partial h}{\partial a}(2^{-1/\varepsilon^3}, \varepsilon) = \lim_{\varepsilon \downarrow 0} (1+\varepsilon^2)2^{-1/\varepsilon} = 0.$$

So, it is possible to see that any representation of **F7** that might involve a maximum of functions cannot have maps of class C^1. Therefore, this function does not satisfy the requirements of our convergence analysis.

Now, let us consider function **F8**, which primarily appeared in a pre-print of [32]. Then, for $Ax \neq 0$, its Hessian can be computed by

$$\nabla^2 f(x) = \frac{1}{2}\left(-\frac{1}{2} g(x)^{-3/2} \nabla g(x) \nabla g(x)^T + g(x)^{-1/2} \nabla^2 g(x)\right),$$

with

$$\nabla g(x) = (x^T A x)^{-1/2} Ax + 2Bx$$

and

$$\nabla^2 g(x) = -(x^T A x)^{-3/2} Ax (Ax)^T + (x^T A x)^{-1/2} A + 2B.$$

Consequently, if one could have a maximum representation of f as in (4.1), then the functions ϕ_i would not be of class C^2, since $\|\nabla^2 g(x)\| \to \infty$ as $\|Ax\| \to 0$.

Fortunately, although those functions do not satisfy the representation hypothesis, when we look at the results obtained by the minimization of **F7** and **F8** (see Figures 23 and 24), we see that this fact is not an obstacle for GraFuS to present a rapid convergence behavior for both functions.

4.4 Failures before reaching the current version of GraFuS

In this brief section, we have the intent to describe a previous idea that did not work out. We believe that this presentation will be helpful for potential researchers, specially because the failures that we have experienced might be fixed.

As we have seen during this chapter, the decomposition of the domain $\mathbb{R}^n$ in two subspaces, namely, U and V spaces, is of great importance for the functioning of GraFuS, but, in the way we have built our algorithm, the identification of those subspaces are not necessary during the execution of the method.

Although this kind of identification is not implemented in the method, it would be desirable to have this procedure inside the method. However, this step presented to be tricker than we were expecting.

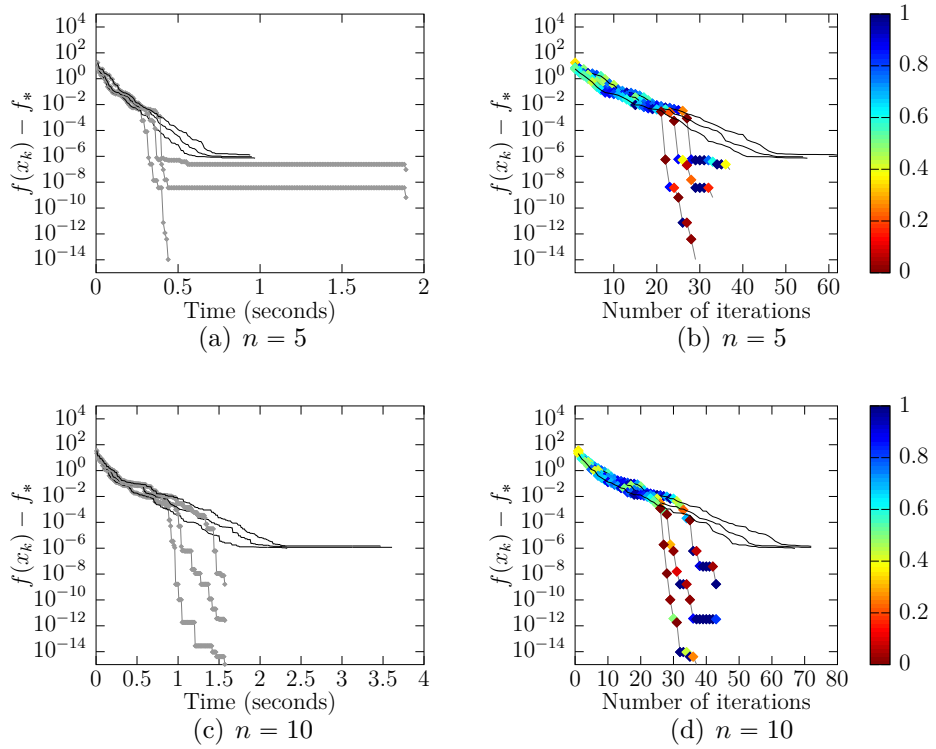


Figure 23 – Medians and quartiles of twenty runs of GS and GraFuS methods for function **F7**. For both number of variables we have $x_* = 0$.

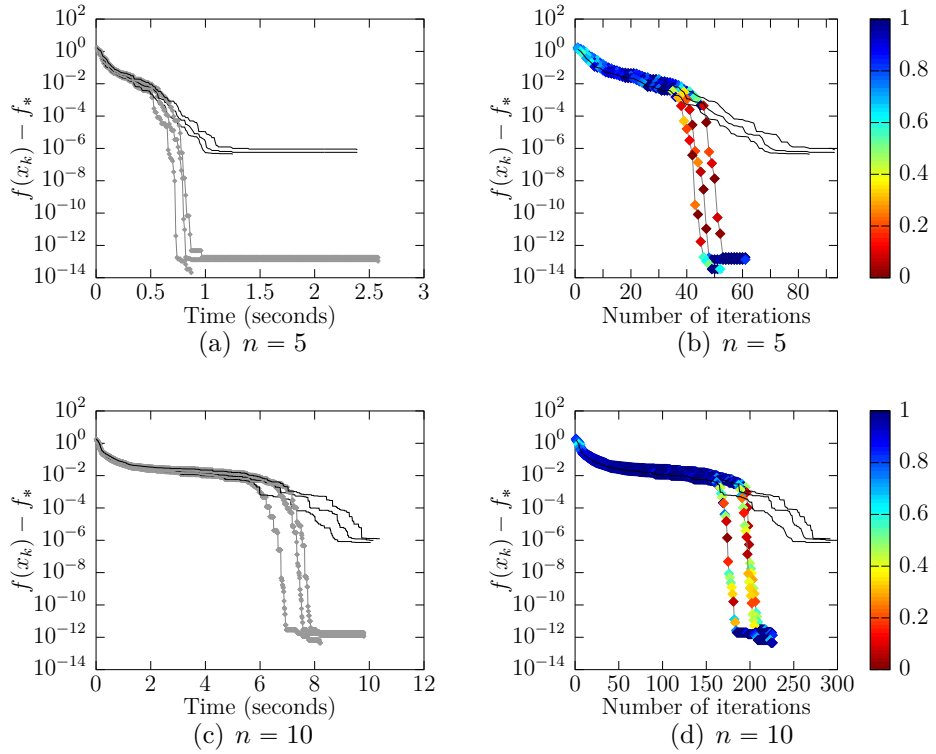


Figure 24 – Medians and quartiles of twenty runs of GS and GraFuS methods for function **F8** with $\delta = 10^{-2}$. For both number of variables we have $x_* = 0$.

First, let us explain the reason why it would be a good idea to identify these subspaces. Because of Theorem 4.3, we see that the only term that prevents us to move superlinearly to the solution of the problem is the distance between the sampled points and the current iterate. Therefore, the parameter $\sigma_l \in [1, 2]$ plays a key role in speeding up the rate of convergence. Hence, if the dimension of $U(x_*)$ is greater than one, it would be desirable to have $\sigma_l > 1$, since it would increase the chance of having a sufficiently small $\tau_{k,\bar{l}_k}$. On the other hand, if the dimension of $U(x_*)$ is zero, it is sufficient to have $\sigma_l = 1$, since in that case, the term $\tau_{k,\bar{l}_k}$ can be of the order of $\|x_k - x_*\|$. More than that, we must have $\sigma_l = 1$, since otherwise, we would prevent the method to satisfy hypothesis $H1$ (see Figure 25).

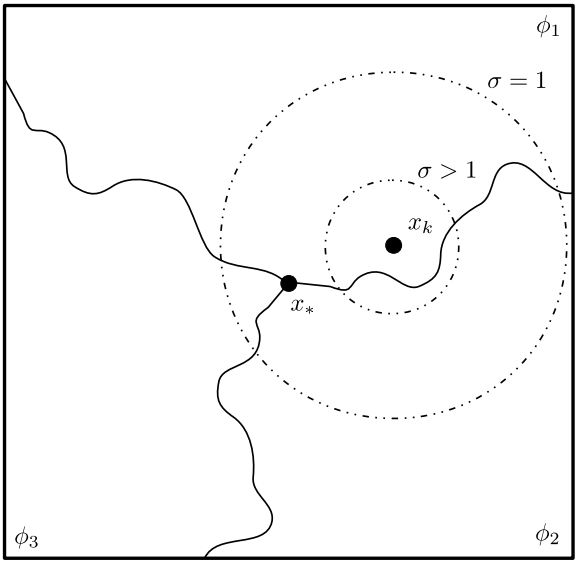


Figure 25 – Representation of how the value σ may interfere in the validity of hypothesis H1. In the figure we represent the regions where each ϕ_i assumes the maximum.

Unfortunately, this kind of identification is not an easy task. An attempt to identify those subspaces at the solution during the execution of the method was done (observing the components of the dual variable λ that were strictly positive and using the related constraints to approximate J_{x_*} and its null space), but we have failed in obtaining a satisfactory result. For this reason, we have established the update of the power of ϵ_k as $\sigma_l = 1 + 0.5^{(l+2)/4} \mod (l + 1, 2)$. With this rule, we try both values of σ_l, i.e., $\sigma_l > 1$ and $\sigma_l = 1$.

A direct implication of this failure for approximating the smooth subspace explains why our attempt to have a sampling algorithm that would generalize the simplified Newton's method [2, Section 14.5] has not succeeded. For this method, we would first move in the nonsmooth subspace, and then, move in the direction of the smooth subspace

related to the function f. Unfortunately, the failure in the identification of those subspaces has prevented us to keep this idea alive.

4.5 Discussion

This chapter presents an implementable algorithm for solving unconstrained nonsmooth and nonconvex optimization problems. Using the ideas of the Gradient Sampling algorithm and taking advantage of some notions developed over the years for the Bundle Method, we were able to produce an algorithm that, in some sense, can be viewed as a generalization of the well established Newton's (quasi-Newton) method.

Additionally, we believe that an important step has been taken in the direction of obtaining a rapid method to minimize nonconvex and nonsmooth functions. It was shown that a rapid move towards the solution is a reliable behavior for some iterations of GraFuS. Moreover, at least for the small set of functions considered in the numerical experiments, one can see that fast moves are not rare and can be expected for a reasonable amount of iterations. However, it must be stressed that the iterations of GraFuS are computationally expensive when compared to GS, and for this reason, the rapid behavior might not be translated to a faster method for some functions. Furthermore, for a number of variables greater than the values considered in the last section, we have experienced good and bad results as well. For example, there is a clear advantage of GraFuS over the GS method for the function **F4**, whereas for **F2**, the results obtained are unsatisfactory (see Figure 26). Nevertheless, if we allow the GS method to work a few more iterations, reasonable results are recovered for the function **F2** (see Figure 27).

The matters of efficiency and applicability of the method are not treated properly in this study, since our aim here was, first, to produce a mathematical theory that would support a rapid convergence to a solution and second, to obtain numerical results that would guarantee a proof-of-concept of the main theoretical results. There are many possibilities of improvements on the algorithm (e.g. different forms of updating the matrices H_k and efficient ways of selecting the sampling radius size without affecting the global convergence) and we hope that future studies explore these possibilities.

Summing up, we end these final remarks with two questions that naturally arise from some of the numerical results obtained in the previous section:

- under which conditions could we establish $\|H_k - H_*\| = O(\|x_k - x_*\|)$ in Theorem 4.3?
- would it be possible to have convergence results with more general assumptions?

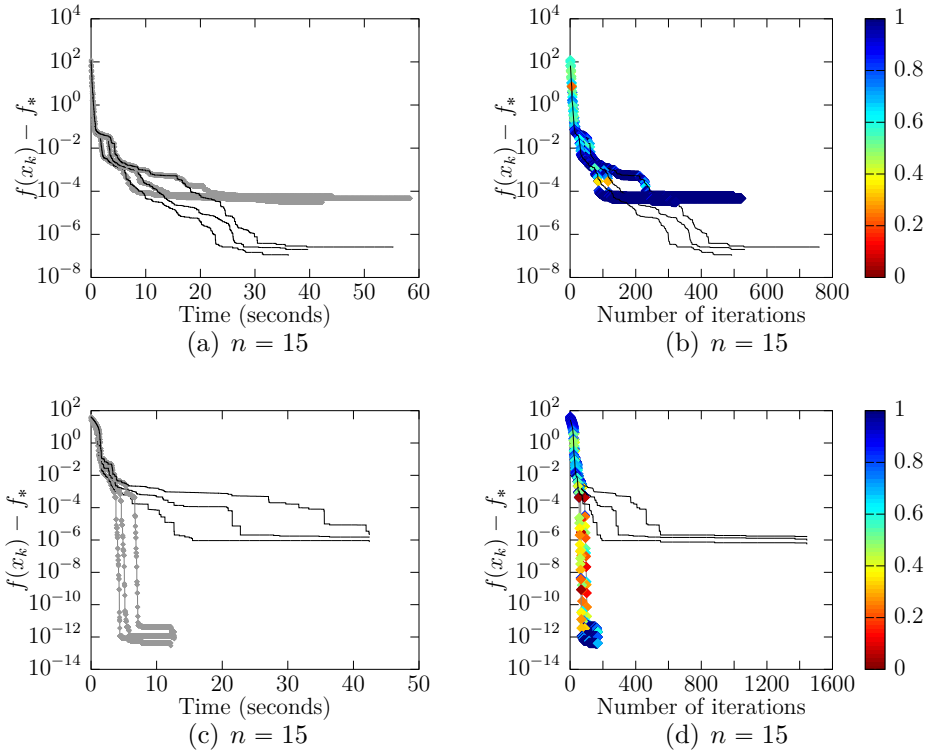


Figure 26 – Medians and quartiles of twenty runs of GS and GraFuS methods for functions **F2** (top) and **F4** (bottom).

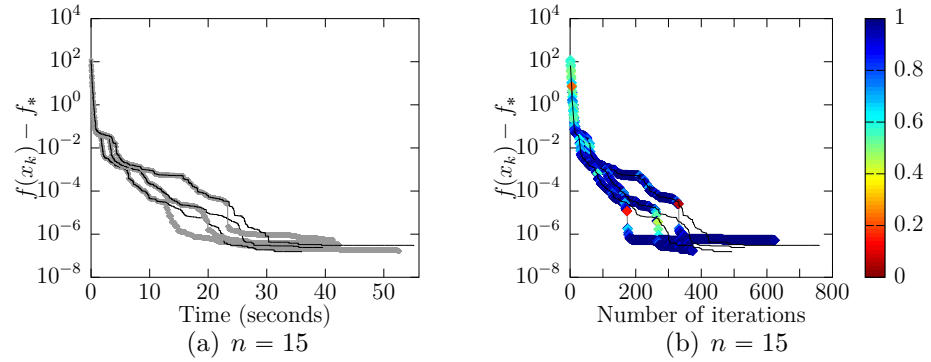


Figure 27 – Medians and quartiles of twenty runs of GS and GraFuS methods for function **F2**, but allowing the GS method to work a few more iterations before we start GraFuS.

5 Final Remarks

In this thesis we have studied sampling techniques to solve unconstrained nonsmooth optimization problems. While bundle methods have been extensively studied over the years, sampling techniques are recent and have opened a new range of possibilities.

The first major contribution of this study is the fact that we were able to avoid the differentiability check step without loosing the global convergence of the preexisting sampling methods. This was possible by doing a perturbation procedure over the search direction or by using a nonmonotone line search. With these alternative steps, we have shortened the gap between the implemented version of GS and the theoretical one.

Further, we have explored the local convergence of the GS method. It was shown that, under special conditions, the GS algorithm extends the linear rate of convergence of the well established Cauchy method. Additionally, we have found theoretical foundation for selecting some key parameters of the GS method, choices that were entirely empirical before.

Finally, we were able to develop a new sampling method that has shown promising practical performance. Using some ideas developed over the years for bundle techniques and taking advantage of some good properties of the GS method, we have created a practical algorithm with rapid local convergence (under special circumstances) to minimize nonsmooth and nonconvex functions, a property that, up to our knowledge, has not been fulfilled yet by any other method in the nonsmooth field.

Bibliography

[1] Michel L. Balinski and Philip Wolfe. *Nondifferentiable Optimization*, volume 3. Math. Programming Studies., USA, 1975.

[2] Joseph F. Bonnans, Jean C. Gilbert, Claude Lemaréchal, and Claudia A. Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Springer-Verlag Berlin Heidelberg, 2nd edition, 2006.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, 2004.

[4] Dmitri Burago, Yuri Burago, and Sergei Ivanov. *A course in metric geometry*, volume 33. American Mathematical Society Providence, 2001.

[5] James V. Burke, Adrian S. Lewis, and Michael L. Overton. Approximating subdifferentials by random sampling of gradients. *Mathematics of Operations Research*, 27(3):567–584, 2002.

[6] James V. Burke, Adrian S. Lewis, and Michael L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2005.

[7] E. Ward Cheney and Allen A. Goldstein. Newton's method for convex programming and Tchebycheff approximation. *Numerische Mathematik*, 1(1):253–268, 1959.

[8] Frank H. Clarke. *Optimization and nonsmooth analysis*, volume 5. SIAM, Montreal, Canada, 1990.

[9] Frank E. Curtis and Michael L. Overton. A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization. *SIAM Journal on Optimization*, 22(2):474–500, 2012.

[10] Frank E. Curtis and Xiaocun Que. An adaptive gradient sampling algorithm for non-smooth optimization. *Optimization Methods and Software*, 28(6):1302–1324, 2013.

[11] Aris Daniilidis, Claudia Sagastizábal, and Mikhail Solodov. Identifying structure of nonsmooth convex functions by the bundle technique. *SIAM Journal on Optimization*, 20(2):820–840, 2009.

[12] Trinh-Minh-Tri Do and Thierry Artières. Regularized bundle methods for convex and non-convex risks. *The Journal of Machine Learning Research*, 13(1):3539–3583, 2012.

[13] D. Dotta, A. S. Silva, and I. C. Decker. Design of power system controllers by nonsmooth, nonconvex optimization. In *Power Energy Society General Meeting, 2009. PES '09. IEEE*, pages 1–7, 2009.

[14] Ding-Zhu Du and Panos M. Pardalos. *Minimax and applications*, volume 4. Springer US, 2013.

[15] Antonio Fuduli, Manlio Gaudioso, and Giovanni Giallombardo. A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. *Optimization Methods and Software*, 19(1):89–102, 2004.

[16] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[17] Andreas Grothey and Kim McKinnon. A superlinearly convergent trust region bundle method. *Report, Department of Mathematics & Statistics, Edinburgh University*, 1998.

[18] Marjo Haarala, Kaisa Miettinen, and Marko M. Mäkelä. New limited memory bundle method for large-scale nonsmooth optimization. *Optimization Methods and Software*, 19(6):673–692, 2004.

[19] Elias S. Helou, Sandra A. Santos, and Lucas E. A. Simões. On the differentiability check in gradient sampling methods. *Optimization Methods and Software*, 31(5):983–1007, 2016.

[20] Elias S. Helou, Sandra A. Santos, and Lucas E. A. Simões. On the local convergence analysis of the gradient sampling method. *http://www.optimization-online.org/DB_HTML/2016/10/5683.html*. *Submitted*, 2016.

[21] Elias S. Helou, Sandra A. Santos, and Lucas E. A. Simões. A second-order information-based gradient and function sampling method for nonconvex, nonsmooth optimization. *http://www.optimization-online.org/DB_HTML/2016/06/5513.html*, 2016.

[22] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex analysis and minimization algorithms I.* Springer Verlag, New York, 1993.

[23] Greg Huber. Gamma function derivation of n-sphere volumes. *The American Mathematical Monthly*, 89(5):301–302, 1982.

[24] James E. Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[25] Krzysztof C. Kiwiel. *Methods of descent for nondifferentiable optimization*, volume 1133. Springer Berlin Heidelberg, 1985.

[26] Krzysztof C. Kiwiel. A tilted cutting plane proximal bundle method for convex nondifferentiable optimization. *Operations research letters*, 10(2):75–81, 1991.

[27] Krzysztof C. Kiwiel. Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization. *SIAM Journal on Optimization*, 18(2):379–388, 2007.

[28] Claude Lemaréchal and Robert Mifflin. Global and superlinear convergence of an algorithm for one-dimensional minimization of convex functions. *Mathematical Programming*, 24(1):241–256, 1982.

[29] Claude Lemaréchal, François Oustry, and Claudia Sagastizábal. The U-Lagrangian of a convex function. *Transactions of the American Mathematical Society*, 352(2):711–729, 2000.

[30] Claude Lemaréchal and Claudia Sagastizábal. Practical aspects of the Moreau–Yosida regularization: Theoretical preliminaries. *SIAM Journal on Optimization*, 7(2):367–385, 1997.

[31] Adrian S. Lewis. Active sets, nonsmoothness, and sensitivity. *SIAM Journal on Optimization*, 13(3):702–725, 2002.

[32] Adrian S. Lewis and Michael L. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141(1-2):135–163, 2013.

[33] Ladislav Lukšan and Jan Vlček. A bundle-Newton method for nonsmooth unconstrained minimization. *Mathematical Programming*, 83(1-3):373–391, 1998.

[34] Marko Mäkelä. Survey of bundle methods for nonsmooth optimization. *Optimization Methods and Software*, 17(1):1–29, 2002.

[35] Pierre Maréchal and Jane J. Ye. Optimizing condition numbers. *SIAM Journal on Optimization*, 20(2):935–947, 2009.

[36] Robert Mifflin and Claudia Sagastizábal. VU-decomposition derivatives for convex max-functions. In Michel Théra and Rainer Tichatschke, editors, *Ill-posed Variational Problems and Regularization Techniques*, volume 477 of *Lecture Notes in Economics and Mathematical Systems*, pages 167–186. Springer Berlin Heidelberg, 1999.

[37] Robert Mifflin and Claudia Sagastizábal. A VU-algorithm for convex minimization. *Mathematical Programming*, 104(2-3):583–608, 2005.

[38] Robert Mifflin and Claudia Sagastizábal. A science fiction story in nonsmooth optimization originating at IIASA. In Martin Grötschel, editor, *Documenta Mathematica Optimization Stories*, pages 291–300. Deutschen Mathematiker-Vereinigung, Bielefeld, 2012.

[39] Jean J. Moreau and Panagiotis D. Panagiotopoulos. *Nonsmooth mechanics and applications*, volume 302. Springer, Vienna, 2014.

[40] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer-Verlag, New York, 2006.

[41] Welington Oliveira and Claudia Sagastizábal. Bundle methods in the XXIst century: A bird's-eye view. *Pesquisa Operacional*, 34(3):647–670, 2014.

[42] Jiři Outrata, Michal Kočvara, and Jochem Zowe. *Nonsmooth approach to optimization problems with equilibrium constraints: theory, applications and numerical results*, volume 28. Kluwer Academic Publishers, The Netherlands, 2013.

[43] Chengbin Peng, Xiaogang Jin, and Meixia Shi. Epidemic threshold and immunization on generalized networks. *Physica A: Statistical Mechanics and its Applications*, 389(3):549–560, 2010.

[44] R. Tyrrell Rockafellar and Roger Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[45] Ekkehard W. Sachs and Stephen M. Sachs. Nonmonotone line searches for optimization algorithms. *Control & Cybernetics*, 40(4), 2011.

[46] Helga Schramm and Jochem Zowe. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2(1):121–152, 1992.

[47] Anders Skajaa. Limited memory BFGS for nonsmooth optimization. *Master's thesis, Courant Institute of Mathematical Science, New York University*, 2010.

[48] Fu-Cheng Wang and Hsuan-Tsung Chen. Design and implementation of fixed-order robust controllers for a proton exchange membrane fuel cell system. *International Journal of Hydrogen Energy*, 34(6):2705–2717, 2009.

[49] Richard L Wheeden and Antoni Zygmund. *Measure and integral: an introduction to real analysis*, volume 308. CRC Press, 2015.

[50] Hongchao Zhang and William W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM Journal on Optimization*, 14(4):1043–1056, 2004.

[51] Jianzhong Zhang, Nae-Heon Kim, and L. Lasdon. An improved successive linear programming algorithm. *Management Science*, 31(10):1312–1331, 1985.

Appendix

APPENDIX A – On the Nonmonotone Line Search

In this complementary part of Chapter 2, we show that if the Zhang and Hager's nonmonotone line search is assumed and we set $\eta_k = C_k - f(x_k)$, then we have that the sequence $\{\eta_k\}$ is summable. The proof is basically the same found in [45] and the little changes are just due to the fact that we do not suppose that a descent direction is obtained in each iteration, a condition that might concern the reader.

Lemma A.1. *Suppose that in the line search defined at Step 4b, we set $\eta_k = C_k - f(x_k)$, where C_k is given by the rules defined at (2.8). Then, if there exists $f_l \in \mathbb{R}$ such that $f(x_k) \geqslant f_l$, for all $k \in \mathbb{N}$, we must have*

$$\sum_{k=0}^{\infty} \eta_k < \infty \quad \textit{and} \quad \eta_k \geqslant 0.$$

Proof. By the line search defined at Step 4b and $\eta_k = C_k - f(x_k)$, we have

$$\begin{aligned} f(x_k + t_k d_k) &< f(x_k) - \beta\alpha_k t_k g_k^T H_k g_k + \eta_k \\ f(x_{k+1}) &< C_k - \beta\alpha_k t_k g_k^T H_k g_k. \end{aligned}$$

Since $\beta\alpha_k t_k g_k^T H_k g_k \geqslant 0$, it holds that $f(x_{k+1}) \leqslant C_k$. So, by (2.8), we have

$$C_{k+1} = \frac{\varrho_k Q_k C_k + f(x_{k+1})}{Q_{k+1}} = \frac{(Q_{k+1} - 1)C_k + f(x_{k+1})}{Q_{k+1}} \leqslant C_k.$$

Therefore, $C_k - C_{k+1} \geqslant 0$. Consequently,

$$\begin{aligned} \eta_{k+1} &= C_{k+1} - f(x_{k+1}) \\ &= \frac{(Q_{k+1} - 1)C_k + f(x_{k+1})}{Q_{k+1}} - f(x_{k+1}) \\ &= \frac{(Q_{k+1} - 1)}{Q_{k+1}} C_k + \frac{(1 - Q_{k+1})}{Q_{k+1}} f(x_{k+1}) \\ &= \frac{(Q_{k+1} - 1)}{Q_{k+1}} (C_k - f(x_{k+1})). \end{aligned}$$

So, noticing that $Q_{k+1} \geqslant 1$ by definition and remembering that $C_k - f(x_{k+1}) \geqslant 0$, we see that $\eta_{k+1} \geqslant 0$.

We only need to prove now that the sequence is summable. Indeed, since $\{Q_k\}$ is a sequence bounded by

$$Q_{k+1} = \varrho_k Q_k + 1 \leqslant \varrho_{\max} Q_k + 1 \leqslant \ldots \leqslant \varrho_{\max}^{k+1} + \ldots + \varrho_{\max} + 1 \leqslant \frac{1}{1 - \varrho_{\max}},$$

we have that

$$\sum_{k=1}^{\infty} \eta_k = \sum_{k=1}^{\infty} \frac{(Q_k - 1)}{Q_k}(C_{k-1} - f(x_k)) \leqslant \frac{\varrho_{\max}}{1 - \varrho_{\max}} \sum_{k=1}^{\infty} \frac{(C_{k-1} - f(x_k))}{Q_k}.$$

Then, taking into account that

$$C_{k-1} - C_k = C_{k-1} - \frac{(Q_k - 1)C_{k-1} + f(x_k)}{Q_k} = \frac{(C_{k-1} - f(x_k))}{Q_k},$$

we obtain, by a telescopic series argument, that

$$\sum_{k=1}^{\infty} \eta_k \leqslant \frac{\varrho_{\max}}{1 - \varrho_{\max}}(C_0 - f_l),$$

which is the desired result. □

APPENDIX B – Enlightening the Hypothesis H3

The aim of this appendix is to show that the hypotheses made in the local convergence subsection are reasonable. More precisely, we take a carefully look at the assumption H3, which seems to be the strongest and unnatural hypothesis. However, we stress that the estrangement of H3 is not in the fact that we are assuming the irrelevance of the trust-region constraint (which due to the random nature of the method seems to be a necessary hypothesis), but on the statement that many of the constraints associated with the sampled points (at least $n-1$, considering $m = 2n$) are inactive.

At first sight, it seems strong to request that only the first $r+1$ constraints of the quadratic programming problem solved in each iteration of GraFuS are active (which is exactly the cardinality of $\mathcal{I}(x_*)$). Although it is acceptable that under a good set of sampled points (hypothesis H1) and close to the solution x_* there will be at least $r+1$ active constraints (hypothesis H2), it is hard to imagine why the quadratic programming problem would not have more active constraints than that (hypothesis H3). Despite this is not an impossible situation, we have the intent to show that even in the case where we have more than $r+1$ active constraints, the results presented in the local convergence subsection do not change. For this purpose, we divide the argumentation in two cases (for both, we assume that H1 and H2 hold and that the trust-region constraint is not playing any role):

A1) The cardinality of $\mathcal{I}(x_*)$ is $n+1$;

A2) The cardinality of $\mathcal{I}(x_*)$ is $r+1$ with $r < n$.

Suppose that A1 holds and let us consider an iterate x_k sufficiently close to x_*. Moreover, assume that the trust-region constraint is irrelevant in the outer and inner iterations k and $\bar{l}_k$, respectively. Then, looking at the optimization problem in (4.12), we see that any additional active constraint will generate an additional active constraint to (4.12) in a way that it will be a linear combination of the first $n+1$ active constraints (by Remark 4.1 and because $\tilde{J}_k$ remains with constant rank in a close neighborhood of x_*). Hence, the solution obtained with or without this additional constraint is the same, which yields that the results presented at the local convergence subsection do not change for this special case.

So, let us consider the more intricate case A2. Moreover, let us assume that there is only one additional constraint, i.e., the number of active constraints is $r+2$ (we

will see that the occurrence of more than one additional constraint will be a straightforward generalization of this simpler case). In other words, we are saying that solving (4.4) is equivalent to minimize

$$\begin{aligned}\min_{(d,z)\in\mathbb{R}^{n+1}} \quad & z+\frac{1}{2}d^TH_kd\\ \text{s.t.} \quad & f\left(x_{k,i}^{\bar{l}_k}\right)+\nabla f\left(x_{k,i}^{\bar{l}_k}\right)^T\left(x_k+d-x_{k,i}^{\bar{l}_k}\right)=z,\ \ 1\leqslant i\leqslant r+2,\end{aligned}$$

where here we assume, as it was done in H1, that rearrangements were done to have the additional constraint as the $(r+2)$-th constraint and that it has the associated sampled point $x_{k,r+2}^{\bar{l}_k}$. Therefore, for an iterate x_k sufficiently close to the solution and a sufficiently small sampling radius, we have, by the continuity of the functions ϕ_i, that only the functions $\phi_1,\ldots,\phi_{r+1}$ can assume the maximum at any sampled point (here, as it was done in the local convergence subsection, we assume without loss of generality that $\mathcal{I}(x_*)=\{1,\ldots,r+1\}$). So, there is $j\in\{1,\ldots,r+1\}$ such that $f(x_{k,r+2}^{\bar{l}_k})=\phi_j(x_{k,r+2}^{\bar{l}_k})$. Consequently, recalling H1, the above minimization problem can be seen as

$$\begin{aligned}\min_{(d,z)\in\mathbb{R}^{n+1}} \quad & z+\frac{1}{2}d^TH_kd\\ \text{s.t.} \quad & \phi_i\left(x_{k,i}^{\bar{l}_k}\right)+\nabla\phi_i\left(x_{k,i}^{\bar{l}_k}\right)^T\left(x_k+d-x_{k,i}^{\bar{l}_k}\right)=z,\ \ 1\leqslant i\leqslant r+1\\ & \phi_j\left(x_{k,r+2}^{\bar{l}_k}\right)+\nabla\phi_j\left(x_{k,r+2}^{\bar{l}_k}\right)^T\left(x_k+d-x_{k,r+2}^{\bar{l}_k}\right)=z,\end{aligned}$$

whose dual optimization problem is written as

$$\begin{aligned}\max_{\lambda\in\mathbb{R}^{r+2}} \quad & \sum_{i=1}^{r+1}\lambda_i\left[\phi_i\left(x_{k,i}^{\bar{l}_k}\right)+\nabla\phi_i\left(x_{k,i}^{\bar{l}_k}\right)^T\left(x_k-x_{k,i}^{\bar{l}_k}\right)\right]\\ & +\lambda_{r+2}\left[\phi_j\left(x_{k,r+2}^{\bar{l}_k}\right)+\nabla\phi_j\left(x_{k,r+2}^{\bar{l}_k}\right)^T\left(x_k-x_{k,r+2}^{\bar{l}_k}\right)\right]\\ & -\frac{1}{2}\left\|\sum_{i=1}^{r+1}\lambda_i\nabla\phi_i(x_{k,i}^{\bar{l}_k})+\lambda_{r+2}\nabla\phi_j(x_{k,r+2}^{\bar{l}_k})\right\|^2_{H_k^{-1}}\\ \text{s.t.} \quad & e^T\lambda=1.\end{aligned}\tag{B.1}$$

Therefore, we can turn this last constrained maximization problem into an unconstrained one by making the following substitution $\lambda_{r+2}=1-\sum_{i=1}^{r+1}\lambda_i$. So, we have

$$\begin{aligned}\max_{\lambda\in\mathbb{R}^{r+1}} \quad & \sum_{i=1}^{r+1}\lambda_i\left[\phi_i\left(x_{k,i}^{\bar{l}_k}\right)+\nabla\phi_i\left(x_{k,i}^{\bar{l}_k}\right)^T\left(x_k-x_{k,i}^{\bar{l}_k}\right)-\phi_j\left(x_{k,r+2}^{\bar{l}_k}\right)\right.\\ & \left.-\nabla\phi_j\left(x_{k,r+2}^{\bar{l}_k}\right)^T\left(x_k-x_{k,r+2}^{\bar{l}_k}\right)\right]+\phi_j\left(x_{k,r+2}^{\bar{l}_k}\right)\\ & +\nabla\phi_j\left(x_{k,r+2}^{\bar{l}_k}\right)^T\left(x_k-x_{k,r+2}^{\bar{l}_k}\right)\\ & -\frac{1}{2}\left\|\sum_{i=1}^{r+1}\lambda_i\left[\nabla\phi_i(x_{k,i}^{\bar{l}_k})-\nabla\phi_j(x_{k,r+2}^{\bar{l}_k})\right]+\nabla\phi_j(x_{k,r+2}^{\bar{l}_k})\right\|^2_{H_k^{-1}}.\end{aligned}$$

Since the above problem is convex, its solution $\overline{\lambda} \in \mathbb{R}^{r+1}$ can be obtained by equaling the derivative of the objective function to the null vector. Consequently, assuming without loss of generality that the function ϕ_j involved in the additional constraint is ϕ_{r+1}, we have

$$\begin{pmatrix} \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right)^T \\ \vdots \\ \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right)^T \end{pmatrix} H_k^{-1} \begin{pmatrix} \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right)^T \\ \vdots \\ \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right)^T \end{pmatrix}^T \overline{\lambda} =$$
$$\begin{pmatrix} \phi_1\left(x_{k,1}^{\bar{l}_k}\right) + \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T\left(x_k - x_{k,1}^{\bar{l}_k}\right) \\ \vdots \\ \phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) + \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T\left(x_k - x_{k,r+1}^{\bar{l}_k}\right) \end{pmatrix}$$
$$- \begin{pmatrix} \phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right) + \nabla\phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right)^T\left(x_k - x_{k,r+2}^{\bar{l}_k}\right) \\ \vdots \\ \phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right) + \nabla\phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right)^T\left(x_k - x_{k,r+2}^{\bar{l}_k}\right) \end{pmatrix}$$
$$- \begin{pmatrix} \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right)^T \\ \vdots \\ \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right)^T \end{pmatrix} H_k^{-1}\nabla\phi_{r+1}\left(x_{k,r+2}^{\bar{l}_k}\right).$$

Now, changing the points $x_{k,r+2}^{\bar{l}_k}$ for $x_{k,r+1}^{\bar{l}_k}$ and redefining

$$\tau_{k,\bar{l}_k} := \max_{1 \leqslant i \leqslant r+2} \left\| x_{k,i}^{\bar{l}_k} - x_k \right\|,$$

we get

$$\begin{pmatrix} \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \\ \vdots \\ \nabla\phi_r\left(x_{k,r}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \\ 0^T \end{pmatrix} H_k^{-1} \begin{pmatrix} \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \\ \vdots \\ \nabla\phi_r\left(x_{k,r}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \\ 0^T \end{pmatrix}^T \overline{\lambda} =$$
$$\begin{pmatrix} \phi_1\left(x_{k,1}^{\bar{l}_k}\right) + \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T\left(x_k - x_{k,1}^{\bar{l}_k}\right) - \phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T\left(x_k - x_{k,r+1}^{\bar{l}_k}\right) \\ \vdots \\ \phi_r\left(x_{k,r}^{\bar{l}_k}\right) + \nabla\phi_r\left(x_{k,r}^{\bar{l}_k}\right)^T\left(x_k - x_{k,r}^{\bar{l}_k}\right) - \phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T\left(x_k - x_{k,r+1}^{\bar{l}_k}\right) \\ 0^T \end{pmatrix}$$
$$- \begin{pmatrix} \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \\ \vdots \\ \nabla\phi_r\left(x_{k,r}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \\ 0^T \end{pmatrix} H_k^{-1}\nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) + O\left(\tau_{k,\bar{l}_k}\right).$$

This last linear system yields

$$\begin{pmatrix} \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \\ \vdots \\ \nabla\phi_r\left(x_{k,r}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \end{pmatrix} H_k^{-1} \begin{pmatrix} \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \\ \vdots \\ \nabla\phi_r\left(x_{k,r}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \end{pmatrix}^T \begin{pmatrix} \overline{\lambda}_1 \\ \vdots \\ \overline{\lambda}_r \end{pmatrix} =$$

$$\begin{pmatrix} \phi_1\left(x_{k,1}^{\bar{l}_k}\right) + \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T\left(x_k - x_{k,1}^{\bar{l}_k}\right) - \phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T\left(x_k - x_{k,r+1}^{\bar{l}_k}\right) \\ \vdots \\ \phi_r\left(x_{k,r}^{\bar{l}_k}\right) + \nabla\phi_r\left(x_{k,r}^{\bar{l}_k}\right)^T\left(x_k - x_{k,r}^{\bar{l}_k}\right) - \phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T\left(x_k - x_{k,r+1}^{\bar{l}_k}\right) \end{pmatrix}$$

$$- \begin{pmatrix} \nabla\phi_1\left(x_{k,1}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \\ \vdots \\ \nabla\phi_r\left(x_{k,r}^{\bar{l}_k}\right)^T - \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \end{pmatrix} H_k^{-1}\nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) + O\left(\tau_{k,\bar{l}_k}\right).$$

Therefore, following the same reasoning used by us to get here, it is possible to see that the first r components of the dual variable $\hat{\lambda} \in \mathbb{R}^{r+1}$ linked to the problem (4.11) must satisfy the last linear system obtained above (not considering the remaining error vector) and, moreover,

$$\hat{\lambda}_{r+1} = 1 - \sum_{i=1}^{r} \hat{\lambda}_i. \tag{B.2}$$

Therefore, considering $\lambda^* \in \mathbb{R}^{r+2}$ the solution of (B.1) and using equation (B.2), we must have

$$\lambda^* = \begin{pmatrix} \hat{\lambda}_1 \\ \vdots \\ \hat{\lambda}_r \\ \lambda^*_{r+1} \\ 1 - \sum_{i=1}^{r} \hat{\lambda}_i - \lambda^*_{r+1} \end{pmatrix} + O\left(\tau_{k,\bar{l}_k}\right) = \begin{pmatrix} \hat{\lambda}_1 \\ \vdots \\ \hat{\lambda}_r \\ \lambda^*_{r+1} \\ \hat{\lambda}_{r+1} - \lambda^*_{r+1} \end{pmatrix} + O\left(\tau_{k,\bar{l}_k}\right).$$

So, to complete our reasoning, notice that since we are supposing that the trust-region constraint does not play any role in the current iteration (i.e. $\omega_{k,\bar{l}_k} = 0$), one can see, by (4.23), that

$$\begin{aligned} d_{k,\bar{l}_k} &= -H_k^{-1}\left[\sum_{i=1}^{r+1} \lambda_i^* \nabla\phi_i(x_{k,i}^{\bar{l}_k}) + \lambda^*_{r+2}\nabla\phi_{r+1}(x_{k,r+2}^{\bar{l}_k})\right] \\ &= -H_k^{-1}\left[\sum_{i=1}^{r} \lambda_i^* \nabla\phi_i(x_{k,i}^{\bar{l}_k}) + \left(\lambda^*_{r+1} + \lambda^*_{r+2}\right)\nabla\phi_{r+1}(x_{k,r+1}^{\bar{l}_k})\right] + O\left(\tau_{k,\bar{l}_k}\right) \\ &= -H_k^{-1}\sum_{i=1}^{r+1} \hat{\lambda}_i \nabla\phi_i(x_{k,i}^{\bar{l}_k}) + O\left(\tau_{k,\bar{l}_k}\right). \end{aligned}$$

Hence, $d_{k,\bar{l}_k}$ is exactly the search direction obtained in (4.11) with an additional error vector. Therefore, the term $O\left(\tau_{k,\bar{l}_k}\right)$ is absorbed by the other error vectors in Theorem 4.3 and the result is still valid.

Finally, remember that we have considered just one additional active constraint to the others $r + 1$ active constraints. However, it is straightforward to see that exactly the same reasoning can be used to prove the result for any other number of additional constraints.