Noname manuscript No.
(will be inserted by the editor)

A Second-Order Information-Based Gradient and Function Sampling Method for Nonconvex, Nonsmooth Optimization

Elias Salomão Helou · Sandra A. Santos · Lucas E. A. Simões

Received: date / Accepted: date

Abstract This paper has the goal to propose a gradient and function sampling method that under special circumstances moves superlinearly to a minimizer of a general class of nonsmooth and nonconvex functions. We present global and local convergence theory with illustrative examples that corroborate and elucidate the theoretical results obtained along the manuscript.

Keywords nonsmooth nonconvex optimization · gradient sampling · local superlinear convergence · global convergence · unconstrained minimization

Mathematics Subject Classification (2000) 65K10 · 90C26

Elias Salomão Helou was partially supported by FAPESP grants 2013/16508-3 and 2013/07375-0 and by CNPq grant 311476/2014-7.

Sandra A. Santos was partially supported by CNPq grant 304032/2010-7, FAPESP grants 2013/05475-7 and 2013/07375-0 and PRONEX Optimization.

Lucas E. A. Simões was supported by FAPESP grant 2013/14615-7.

Elias Salomão Helou
Institute of Mathematical Sciences and Computation, University of São Paulo.
São Carlos - SP, Brazil.
E-mail: elias@icmc.usp.br

Sandra A. Santos
Department of Applied Mathematics, University of Campinas.
Campinas - SP, Brazil.
E-mail: sandra@ime.unicamp.br

Lucas E. A. Simões
Department of Applied Mathematics, University of Campinas.
Campinas - SP, Brazil.
E-mail: simoes.lea@gmail.com

1 Introduction

Problems involving continuous nonsmooth functions arise in many fields of science [30,37,39], 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 [34,36], which has been successfully solved by well established optimization algorithms known as *Bundle Methods* [1,21,29]. However, a significant amount of problems involve minimizations of nonsmooth functions that are also nonconvex [10,11], a property that usually introduces an undesirable complexity to the implementation of the aforementioned method.

Recently, an algorithm known as *Gradient Sampling* (GS) [5,22] has gained attention for providing good alternatives to the difficulties that Bundle Methods need to deal with if the function is not convex (see [29,35] 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.

Since we can interpret the GS algorithm as a generalization of the steepest descent method, it is reasonable to think that in the best case scenario, the method would have linear local convergence (for the best of our knowledge, there is no local convergence proof for GS). Although the method has shown to be robust, presenting good numerical results even for challenging problems, its cost can be an obstacle, specially if we take into account the expectation over the rate of convergence. Therefore, this leads us 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 manuscript 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 [8, 9], 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 function, 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 [16,28], but still keeps the GS facilities to handle nonconvex functions. This last characteristic is in agreement with Kiwiel's expectation [22]

> "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 [26,31]. Roughly speaking, it tries to behave like the cutting-plane methods [13,20] into 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 overshadowed by the cost of evaluating the gradient.

Finally, 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 [23,24,25,32]). The pursuit for such an algorithm has raised efforts of many researchers (an enlightening review can be found in [33]) and up to our knowledge there is no method in the literature that fulfills those features. 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.

For clarity, before we start to expose the main ideas of this study, we present some notations that appear along this manuscript:

- $\operatorname{co}\mathcal{X}$ is the convex hull of $\mathcal{X}$;
- $\operatorname{cl}\mathcal{X}$ is the closure of $\mathcal{X}$;
- $|\mathcal{X}|$ is the cardinality of $\mathcal{X}$;
- $\mathcal{B}(x,r)$ is the Euclidean open ball with center at x and radius r;
- $\|\cdot\|$ is any norm in $\mathbb{R}^n$;
- $\|x\|_H := \sqrt{x^T H x}$, for any symmetric positive definite matrix H;
- e is a vector with ones in all entries;
- $\operatorname{dist}_H(x,\mathcal{C}) := \inf_{y\in\mathcal{C}} \|y-x\|_H$ with $x \in \mathbb{R}^n$ and $\mathcal{C} \subset \mathbb{R}^n$ a nonempty set;
- $\mathcal{P}[x \in \mathcal{X}]$ is the probability of x to be in $\mathcal{X}$, whereas $\mathcal{P}[x \in \mathcal{X} \mid x \in \mathcal{Y}]$ is the conditional probability of x to be in $\mathcal{X}$ given that $x \in \mathcal{Y}$.

2 Basic concepts and the GS algorithm

The GS method has the goal of solving the following unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz function, continuously differentiable in an open dense subset $\mathcal{D} \subset \mathbb{R}^n$. The function f is not necessarily convex.

For a map with the properties above, it is possible to define the Clarke's subdifferential set for f at x [6, 7]. This set can be interpreted as a generalization of the gradient for nonsmooth functions.

Definition 1 (Subdifferential set, subgradient, stationary point) The set given by

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

is called the Clarke's subdifferential set for f at x and any $v \in \overline{\partial} f(x)$ is known as a subgradient of f at x. Moreover, if $0 \in \overline{\partial} f(x)$, then we say that x is a stationary point for f.

A set that fits best with the idea of sampled points and is more general than the previous one can be defined [15].

Definition 2 (ϵ-Subdifferential set, ϵ-subgradient, ϵ-stationary point) The ϵ-subdifferential set for 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.

With a great importance for our study, we present the generalized directional derivative for the function f [6].

Definition 3 (Generalized directional derivative) The generalized directional derivative of a continuous locally Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$ at x in the direction $v \in \mathbb{R}^n$ is given by

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

Finally, it is possible to link Definition 3 with the subdifferential set. Indeed, the following relation holds [6]

$$f^\circ(x; v) = \max\{s^T v \mid s \in \overline{\partial} f(x)\}.$$

With the above sets in mind, one can interpret the sampled points used in GS method as an attempt to approximate the ϵ-subdifferential set of f at x [4, Theorem 2.1].

For a more complete idea of the GS functioning, we present an algorithm that embraces several GS methods recently developed (see Algorithm 1). For the matrices H_k that appear inside Algorithm 1, we assume a property that is commonly used in quasi-Newton techniques [9].

Algorithm 1: A general algorithmic framework for GS methods.

```
Step 0. Set $k = 0$, $x_0 \in \mathcal{D}$, $m \in \mathbb{N}$ with $m \geq n + 1$, fixed real numbers $\nu_0, \epsilon_0$,
        $\nu_{\text{opt}}, \epsilon_{\text{opt}}$, $\vartheta > 0$ 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.[1]
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 $\lambda_k$ solves

            $\min_{\lambda} \ \frac{1}{2} \lambda^T \tilde{G}_k^T H_k^{-1} \tilde{G}_k \lambda$
            $\text{s.t.} \ e^T \lambda = 1, \ \lambda \geq 0$

        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 $\min\{\|\tilde{g}_k\|_2, \|\tilde{g}_k\|_{H_k}\} < \nu_k$, 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 $\alpha_k \in \{1, \vartheta/\|\tilde{g}_k\|\}$.
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
        $x_{k+1} \in \mathcal{B}(x_k + t_k d_k, \min\{t_k, \epsilon_k\}\|d_k\|_2)$, where the following holds
        $f(x_{k+1}) \leq f(x_k) - \beta \alpha_k t_k \tilde{g}_k^T H_k \tilde{g}_k$.
Step 6. Set $k \leftarrow k + 1$ and go back to Step 1.
```

Assumption 1 *For every $k \in \mathbb{N}$, the matrix $H_k \in \mathbb{R}^{n \times n}$ is symmetric positive definite and there exist positive real numbers $\underline{\varsigma}$ and $\overline{\varsigma}$ such that*

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

Consequently, considering $H_k \equiv I$, we obtain the standard GS method (if $\alpha_k \equiv 1/\|g_k\|_2$) and the GS method with non-normalized direction (if $\alpha_k \equiv 1$). Updating H_k with limited memory LBFGS techniques, one can recover the method introduced in [9].

It is possible to show that if $x_k \in \mathcal{D}$, then the vector d_k used at Step 4 is a descent direction for f at x_k [5], which evinces the importance of Step 5 for the finiteness of the line search procedure (in fact, this procedure is a delicate

[1] Notice that since $\mathcal{D}$ is an open dense subset of $\mathbb{R}^n$, it is easy to guarantee that all the sampled points have probability one to be in $\mathcal{D}$. Hence, with probability one, the corresponding gradients are well defined. This will also be true for our proposed method, presented in the next section.

matter [18]). Moreover, given the random nature of the method, convergence with probability one results are always established [22].

Once we have presented some basic notions about nonsmooth functions and the GS methods, we are able to proceed with the main ideas of this paper.

3 Motivation and the new algorithm

Henceforward, we will be interested in solving a class of problems more structured than (1). Let us consider the minimax optimization problem

$$\min_{x\in\mathbb{R}^n} \left(f(x) := \max_{1\le i\le p} \{\phi_i(x)\} \right), \tag{2}$$

where the functions $\phi_i : \mathbb{R}^n \to \mathbb{R}$ are all of class C^2, but they are not necessarily known. Here, 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 [12] and references therein).

3.1 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

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

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

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

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

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

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

$$\min_{d\in\mathbb{R}^n} \max_{0\leq i\leq m} \left\{ f(x_k) + \nabla f(x_{k,i})^T d + \frac{1}{2} d^T H_k d \right\}. \tag{3}$$

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

$$\min_{d\in\mathbb{R}^n} \max_{1\leq i\leq m} \left\{ f(x_{k,i}) + \nabla f(x_{k,i})^T (x_k - x_{k,i}) + \nabla f(x_{k,i})^T d + \frac{1}{2} d^T H_k d \right\}, \tag{4}$$

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), 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 Step 5 is no longer necessary.

Unfortunately, this new quadratic programming 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 (3). Therefore, to have an algorithm that uses the function values at all sampled points, we must overcome this issue.

3.2 New algorithm

In order to surpass the difficulty of not having a descent direction under a bad sample, we have replaced the Armijo's line search by a trust-region procedure. Besides, to have a smooth problem to solve, instead of dealing with (4), we solve at each iteration the following quadratic optimization problem

$$\begin{aligned} \min_{(d,z)} \quad & z + \frac{1}{2} d^T H_k d \\ \text{s.t.} \quad & \tilde{f}_k + G_k^T d \leq ze \\ & \|d\|_\infty \leq \Delta_k, \end{aligned} \tag{5}$$

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 $\|d\|_\infty \leq \Delta_k$ stands for the trust-region constraint, for some $\Delta_k > 0$. Consequently, its dual optimization problem, after a changing of variables, can be viewed as

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

With these modifications in mind, we introduce the proposed algorithm (Algorithm 2), also referred 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;
- The role played by the exponent σ 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.

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

```
Step 0. Set $k, l = 0$, $x_0 \in \mathbb{R}^n$, $m \in \mathbb{N}$ with $m \geq n+1$ and fixed real numbers
        $\gamma_\epsilon, \gamma_\Delta > 0$, $0 < \nu_0, \theta, \rho < 1$, $0 \leq \nu_{\text{opt}} < \nu_0$ 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 $\sigma$ as any real number in $[1,2]$ and choose
            $\left\{x_{k,1}^l, \ldots, x_{k,m}^l\right\} \subset \mathcal{B}(x_k, \gamma_\epsilon \epsilon_{k,l}^\sigma)$
        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, (5) and its dual
        problem, where $H_k \in \mathbb{R}^{n \times n}$ is a symmetric and positive definite matrix.
Step 3. If $\nu_k \leq \nu_{\text{opt}}$, then STOP! Otherwise, compute
            $\text{Ared}_{k,l} := f(x_k) - f(x_k + d_{k,l})$
        and
            $\text{Pred}_{k,l} := \max_i \left\{f(x_{k,i}^l) + \nabla f(x_{k,i}^l)^T (x_k - x_{k,i}^l)\right\} - \left(z_{k,l} + \frac{1}{2} d_{k,l}^T H_k d_{k,l}\right).$
Step 4. If $\text{Ared}_{k,l} \leq \rho \text{Pred}_{k,l}$, then set $\Delta_{k,l+1} = \theta \Delta_{k,l}$, $\epsilon_{k,l+1} = \theta \epsilon_{k,l}$,
        $l \leftarrow l+1$ and go back to Step 1. Otherwise, set $x_{k+1} = x_k + d_{k,l}$ 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.
```

Glossary of Notation	
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: exponent related to the sampling size

4 Convergence

Before we proceed with the convergence analysis, we should state a property for the functions ϕ_i that define f. It is a common assumption when we are dealing with nonsmooth functions of the kind defined in (2), cf. [31]. Therefore, considering that $\mathcal{I}(x) := \{i \mid \phi_i(x) = f(x)\}$, the required hypothesis follows.

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

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

In addition, from now on, we also suppose that the matrices H_k in GraFuS satisfy Assumption 1.

Remark 1 It is worth pointing out that Assumption 2 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 $|\mathcal{I}(x)| \geq 2$ (the proof is provided in Lemma 1 below). This association will be of great importance for both the global and the local convergence results.

4.1 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 1 *Let x be any point in $\mathbb{R}^n$ and j be any fixed index in $\mathcal{I}(x)$. Then, $|\mathcal{I}(x)| \leq n+1$. 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 $\operatorname{int}(\mathcal{C}_j(x,\epsilon)) \neq \emptyset$, for which $x \notin \mathcal{C}_j(x,\epsilon)$ and*

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

Proof Let us prove first that $|\mathcal{I}(x)| \leq n+1$. If $|\mathcal{I}(x)| = 1$, the statement trivially holds. Therefore, we assume that $|\mathcal{I}(x)| \geq 2$. Besides, we suppose without any loss of generality that $\mathcal{I}(x) = \{1, \dots, r\}$. Then, let $\alpha_2, \dots, \alpha_r \in \mathbb{R}$ 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 2, we have $\alpha_2=\ldots=\alpha_r=0$. Consequently,

$$\mathcal{A}:=\{\nabla\phi_i(x)-\nabla\phi_1(x)\}_{i\in\mathcal{I}(x)\backslash\{1\}}$$

forms a linearly independent set. So, $|\mathcal{A}|\leq n$, which implies that $|\mathcal{I}(x)|\leq n+1$.

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 $|\mathcal{I}(x)|\geq 2$ and $\mathcal{I}(x)=\{1,\ldots,r\}$. By Assumption 2, 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 $|\mathcal{I}(x)|\leq n+1$ and notice that the set formed by the vectors v_j's is a linearly independent set). Thus, it is possible to find a unitary $d_j\in\mathbb{R}^n$ such that $v_j^Td_j>0$ and

$$v_i^Td_j=0,\;\; i\neq j \;\;\text{with}\;\; i\in\mathcal{I}(x).^2$$

Consequently, it follows that $\nabla\phi_j(x)^Td_j>\nabla\phi_s(x)^Td_j$ and

$$\nabla\phi_i(x)^Td_j=\nabla\phi_s(x)^Td_j,\;\; i\neq j\;\;\text{with}\;\; i\in\mathcal{I}(x).$$

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

$$w_j\in\mathcal{B}\left(0,\delta\right)\subset\mathbb{R}^n,\text{ where }\delta:=\min_{\substack{i\in\mathcal{I}(x)\\ i\neq j}}\left\{\frac{[\nabla\phi_j(x)-\nabla\phi_i(x)]^Td_j}{2\|\nabla\phi_j(x)-\nabla\phi_i(x)\|}\right\}>0,$$

[2] 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

$$\begin{aligned}\frac{\phi_j(x+\epsilon(d_j+w_j)) - \phi_i(x+\epsilon(d_j+w_j))}{\epsilon} &= [\nabla\phi_j(x) - \nabla\phi_i(x)]^T d_j \\ &\quad + [\nabla\phi_j(x) - \nabla\phi_i(x)]^T w_j + O(\epsilon) \\ &\geq [\nabla\phi_j(x) - \nabla\phi_i(x)]^T d_j \\ &\quad - \|\nabla\phi_j(x) - \nabla\phi_i(x)\| \|w_j\| + O(\epsilon) \\ &> \frac{[\nabla\phi_j(x) - \nabla\phi_i(x)]^T d_j}{2} + O(\epsilon).\end{aligned}$$

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 $\epsilon \geq r$, since in this case we have $\mathcal{B}(x,r) \subset \mathcal{B}(x,\epsilon)$), the following set is not empty

$$\mathcal{S}_j(x,\epsilon) := \operatorname{int}\left\{ y \in \mathcal{B}(x,\epsilon) \;\Big|\; \phi_j(y) > \max_{\substack{1 \leq i \leq p \\ i \neq j}} \phi_i(y) \right\}, \;\; j \in \mathcal{I}(x). \qquad (6)$$

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 [40]).

Lemma 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* (6) *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 $j \in \mathcal{I}(\overline{x})$*, we have* $i \in \{1, \ldots, m\}$ *that implies* $x^l_{k,i} \in \mathcal{S}_j(\overline{x}, \overline{\epsilon})$*;*

c) for all $i \in \{1, \ldots, m\}$*, there exists* $j \in \mathcal{I}(\overline{x})$ *that implies* $x^l_{k,i} \in \mathcal{S}_j(\overline{x}, \overline{\epsilon})$*,*

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

$$\phi_j(x) > \max_{\substack{1 \le i \le p \\ i \notin \mathcal{I}(\overline{x})}} \phi_i(x), \ \text{ for all } j \in \mathcal{I}(\overline{x}).$$

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

$$s^T v \le -\tau, \ \ \forall s \in \overline{\partial} f(\overline{x}).$$

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 $f^\circ(\overline{x}; v) \le -\tau$. 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

$$f(x + \Delta v) - f(x) < -\frac{\tau}{2}\Delta. \tag{7}$$

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

$$\begin{aligned} f(x_k) &= \max_{i \in \mathcal{I}(\overline{x})} \{\phi_i(x_k)\} \\ &= \max_{1 \le i \le m} \{f(x^l_{k,i}) + \nabla f(x^l_{k,i})^T (x_k - x^l_{k,i})\} + o(\Delta_{k,l}) \\ &\quad \text{(notice that } x^l_{k,i} \in \mathcal{B}(x_k, M \cdot \Delta_{k,l})) \end{aligned}$$

and

$$\begin{aligned} f(x_k + d_{k,l}) &= \max_{i \in \mathcal{I}(\overline{x})} \{\phi_i(x_k + d_{k,l})\} \\ &= \max_{1 \le i \le m} \{f(x^l_{k,i}) + \nabla f(x^l_{k,i})^T (x_k + d_{k,l} - x^l_{k,i})\} + o(\Delta_{k,l}) \\ &\quad \text{(notice that } x^l_{k,i} \in \mathcal{B}(x_k, M \cdot \Delta_{k,l}) \text{ and that } \|d_{k,l}\|_\infty \le \Delta_{k,l}). \end{aligned}$$

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

$$\hat{z} = \max_{1 \le i \le m} \{ f(x_{k,i}^l) + \nabla f(x_{k,i}^l)^T (x_k + \Delta_{k,l} v - x_{k,i}^l) \}.$$

Therefore, since $(d_{k,l}, z_{k,l})$ is the solution of the quadratic programming problem at Step 2, we have that $z_{k,l} \le \hat{z} + o(\Delta_{k,l})$, and hence,

$$\text{Pred}_{k,l} \ge \max_i \{ f(x_{k,i}^l) + \nabla f(x_{k,i}^l)^T (x_k - x_{k,i}^l) \} - \left(\hat{z} + \frac{\Delta_{k,l}^2}{2} v^T H_k v \right) + o(\Delta_{k,l}).$$

Consequently, it yields that

$$\begin{aligned} \text{Pred}_{k,l} &\ge f(x_k) - f(x_k + \Delta_{k,l} v) + o(\Delta_{k,l}) \\ &> \frac{\tau}{2} \Delta_{k,l} + o(\Delta_{k,l}), \end{aligned}$$

where the last inequality comes from (7). 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 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 $l \le \bar{l}$.*

Proof Let us assume, for contradiction, that such $\bar{l}$ does not exist, i.e., $l \to \infty$ at the iteration k. Consequently, we must have that $\text{Ared}_{k,l} \le \rho \text{Pred}_{k,l}$, for all $l \in \mathbb{N}$. Additionally, by the way we have designed our algorithm, we see that

$$\epsilon_{k,l} = \frac{1}{\gamma_\Delta} \Delta_{k,l}, \ \ \forall k, l \in \mathbb{N},$$

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 2, it is straightforward to see that at some $\tilde{n} \in \mathbb{N}$, if $l \ge \tilde{n}$, then hypotheses *i*) and *ii*) of Lemma 2 are valid. Moreover, considering $\overline{\epsilon} := \gamma_\epsilon \epsilon_{k,l}^\sigma$ 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 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 $\mathcal{C}_j(x_k, \gamma_\epsilon \epsilon^\sigma_{k,l})$ in the proof of Lemma 1, we have that (for $\gamma_\epsilon \epsilon^\sigma_{k,l}$ sufficiently small) $\mathcal{B}^{k,l}_j \subset \mathcal{C}_j(x_k, \gamma_\epsilon \epsilon^\sigma_{k,l})$, where

$$\mathcal{B}^{k,l}_j := \mathcal{B}\left(x_k + \frac{\gamma_\epsilon \epsilon^\sigma_{k,l}}{4} d_j, \frac{\gamma_\epsilon \epsilon^\sigma_{k,l}}{8} \min_{\substack{i \in \mathcal{I}(x_k) \\ i \neq j}} \left\{ \frac{[\nabla \phi_j(x_k) - \nabla \phi_i(x_k)]^T d_j}{2\|\nabla \phi_j(x_k) - \nabla \phi_i(x_k)\|} \right\} \right).$$

Consequently, the volume of $\mathcal{B}^{k,l}_j$ in $\mathbb{R}^n$ is given by

$$\mathrm{Vol}\left(\mathcal{B}^{k,l}_j\right) = \frac{\pi^{n/2}}{\Gamma(n/2+1)} \left(\min_{\substack{i \in \mathcal{I}(x_k) \\ i \neq j}} \left\{ \frac{[\nabla \phi_j(x_k) - \nabla \phi_i(x_k)]^T d_j}{2\|\nabla \phi_j(x_k) - \nabla \phi_i(x_k)\|} \right\} \right)^n \left(\frac{\gamma_\epsilon \epsilon^\sigma_{k,l}}{8} \right)^n,$$

where Γ is the Gamma function [19]. On the other hand, it follows that

$$\mathrm{Vol}(\mathcal{B}(x_k, \gamma_\epsilon \epsilon^\sigma_{k,l})) = \frac{\pi^{n/2}}{\Gamma(n/2+1)} \left(\gamma_\epsilon \epsilon^\sigma_{k,l}\right)^n.$$

Therefore, since the sampled points are chosen in $\mathcal{B}(x_k, \gamma_\epsilon \epsilon^\sigma_{k,l})$ and

$$\mathcal{B}^{k,l}_j \subset \mathcal{C}_j(x_k, \gamma_\epsilon \epsilon^\sigma_{k,l}) \subset \mathcal{S}_j(x_k, \gamma_\epsilon \epsilon^\sigma_{k,l}),$$

we must have, for all $i \in \{1, \ldots, m\}$, that the conditional probability

$$\mathcal{P}(x^l_{k,i} \in \mathcal{S}_j(x_k, \gamma_\epsilon \epsilon^\sigma_{k,l}) \mid x^l_{k,i} \in \mathcal{B}(x_k, \gamma_\epsilon \epsilon^\sigma_{k,l})) = \frac{\mathrm{Vol}(\mathcal{S}_j(x_k, \gamma_\epsilon \epsilon^\sigma_{k,l}))}{\mathrm{Vol}(\mathcal{B}(x_k, \gamma_\epsilon \epsilon^\sigma_{k,l}))}$$

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 *Let us consider the GraFuS algorithm. If* $Pred_{k,\bar{l}_k}/\Delta_{k,\bar{l}_k} \to 0$ *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 (5) satisfies the Slater's condition. Indeed, if one considers $d_k = 0$ and $z_k = \max\{\tilde{f}_k\} + 1$ in (5), 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

$$\begin{aligned} z_{k,\bar{l}_k} + \frac{1}{2} d_{k,\bar{l}_k}^T H_k d_{k,\bar{l}_k} = \; & \lambda_{k,\bar{l}_k}^T \tilde{f}_{k,\bar{l}_k} \\ & - \frac{1}{2}\left(G_{k,\bar{l}_k}\lambda_{k,\bar{l}_k} + \omega_{k,\bar{l}_k}\right)^T H_k^{-1}\left(G_{k,\bar{l}_k}\lambda_{k,\bar{l}_k} + \omega_{k,\bar{l}_k}\right) \\ & - \varDelta_{k,\bar{l}_k}\|\omega_{k,\bar{l}_k}\|_1. \end{aligned}$$

Thus, defining

$$\alpha_k := \frac{1}{2}\left(G_{k,\bar{l}_k}\lambda_{k,\bar{l}_k} + \omega_{k,\bar{l}_k}\right)^T H_k^{-1}\left(G_{k,\bar{l}_k}\lambda_{k,\bar{l}_k} + \omega_{k,\bar{l}_k}\right) + \varDelta_{k,\bar{l}_k}\|\omega_{k,\bar{l}_k}\|_1, \quad (8)$$

it yields

$$\begin{aligned} \lambda_{k,\bar{l}_k}^T \tilde{f}_{k,\bar{l}_k} - \alpha_k = z_{k,\bar{l}_k} + \frac{1}{2} d_{k,\bar{l}_k}^T H_k d_{k,\bar{l}_k} & \Rightarrow \alpha_k = \lambda_{k,\bar{l}_k}^T \tilde{f}_{k,\bar{l}_k} \\ & \qquad - \left(z_{k,\bar{l}_k} + \frac{1}{2} d_{k,\bar{l}_k}^T H_k d_{k,\bar{l}_k}\right) \\ & \Rightarrow \alpha_k \le \mathrm{Pred}_{k,\bar{l}_k} \\ & \quad \text{(since } \lambda_{k,\bar{l}_k} \ge 0 \text{ and } e^T\lambda_{k,\bar{l}_k} = 1\text{)} \\ & \Rightarrow \frac{\alpha_k}{\varDelta_{k,\bar{l}_k}} \le \frac{\mathrm{Pred}_{k,\bar{l}_k}}{\varDelta_{k,\bar{l}_k}} \\ & \Rightarrow \frac{\alpha_k}{\varDelta_{k,\bar{l}_k}} \to 0. \end{aligned}$$

Consequently, by Assumption 1 and (8), we obtain $\|G_{k,\bar{l}_k}\lambda_{k,\bar{l}_k}\| \to 0$. □

Now, we present the main goal of this subsection. Below, we prove the global convergence (with probability one) of the proposed algorithm.

Theorem 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 $\overline{\epsilon} > 0$ such that $\epsilon_{k,\bar{l}_k} \ge \overline{\epsilon}$ 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 $\epsilon_{k,\bar{l}_k} \le \nu_k$, 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 $\nu_k \ge \overline{\nu}$, for all $k \in \mathbb{N}$. Additionally, we claim that there exists $\mu > 0$ such that $\Delta_{k,\bar{l}_k}\mu \leq \mathrm{Pred}_{k,\bar{l}_k}$, 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

$$\mathrm{Pred}_{k,\bar{l}_k}/\Delta_{k,\bar{l}_k} \underset{k\in\tilde{\mathcal{K}}}{\rightarrow} 0.$$

However, by Lemma 4, it would yield that

$$\|G_{k,\bar{l}_k}\lambda_{k,\bar{l}_k}\| \underset{k\in\tilde{\mathcal{K}}}{\rightarrow} 0.$$

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 $\Delta_{k,\bar{l}_k}\mu \leq \mathrm{Pred}_{k,\bar{l}_k}$, for all $k \in \mathbb{N}$. Moreover, since

$$\epsilon_{k,l} = \frac{1}{\gamma_\Delta}\Delta_{k,l}, \;\; \forall k, l \in \mathbb{N},$$

it yields that $\Delta_{k,\bar{l}_k} \geq \gamma_\Delta \bar{\epsilon}$, for all $k \in \mathcal{K}_1$. Consequently,

$$\mathrm{Ared}_{k,\bar{l}_k} > \rho \mathrm{Pred}_{k,\bar{l}_k}, \;\; \forall k \in \mathcal{K}_1 \Rightarrow f(x_k) - f(x_{k+1}) > \rho\mu\gamma_\Delta\bar{\epsilon}, \;\; \forall k \in \mathcal{K}_1. \quad (9)$$

Now, since $\{x_k\}$ is a bounded sequence, there must exist an infinite set of indices $\mathcal{K}_2 \subset \mathcal{K}_1$ such that

$$x_k \underset{k\in\mathcal{K}_2}{\rightarrow} \hat{x}, \;\; \text{for some } \hat{x} \in \mathbb{R}^n.$$

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

$$\sum_{k\in\mathcal{K}_2} (f(x_k) - f(x_{k+1})) \leq \sum_{k\in\mathcal{K}_2} \left(f(x_k) - f\left(x_{s_{\mathcal{K}_2}(k)}\right)\right) = f(x_w) - f(\hat{x}) < \infty,$$

with $w \in \mathbb{N}$ being the first index in $\mathcal{K}_2$. However, this is a condition that goes against (9). 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

$$x_k \underset{k\in\mathcal{K}}{\rightarrow} \overline{x}.$$

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 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 $k \geq k'$ we have

$$\Delta_{k,l'} = \tilde{\Delta} := \left(\theta^{l'}\right) \gamma_\Delta \nu_k < \overline{\Delta} \text{ and } \epsilon_{k,l'} = \tilde{\epsilon} := \left(\theta^{l'}\right) \nu_k = \frac{1}{\gamma_\Delta} \tilde{\Delta}.$$

Moreover, since $\overline{x}$ is a cluster point for the iteration sequence, we can find $\hat{k} \geq k'$ such that for all $k \geq \hat{k}$ and $k \in \mathcal{K}$, we have $x_k \in \mathcal{B}(\overline{x}, \min\{\gamma_\epsilon \tilde{\epsilon}, \overline{\delta}\}/4)$. So, it yields that for all $j \in \mathcal{I}(\overline{x})$ we have

$$x_k \in \mathcal{B}(\overline{x}, \min\{\gamma_\epsilon \tilde{\epsilon}, \overline{\delta}\}/4) \text{ and } \mathcal{S}_j(\overline{x}, \min\{\gamma_\epsilon \tilde{\epsilon}^\sigma, \overline{\delta}\}/4) \subset \mathcal{B}(x_k, (\gamma_\epsilon/\gamma_\Delta)\tilde{\Delta}).$$

Therefore, the hypotheses *i*), *ii*) and *iii*) item *a*) of Lemma 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 $k \geq \hat{k}$ and $l = l'$. However, this is an event with probability zero to happen, since the sets $\mathcal{S}_j(\overline{x}, \min\{\gamma_\epsilon \tilde{\epsilon}^\sigma, \overline{\delta}\}/4)$ 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 [22, 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 Local convergence

In this subsection our efforts will be focused in enlightening the role played by the quadratic programming problem (5). 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 (2). Also, assume that $\mathcal{I}(x_*) = \{1, \ldots, r+1\}$, for some $r \leq n$. Therefore, consider any $x_k \in \mathbb{R}^n$ and the sampled points $x_{k,1}^{\bar{l}_k}, \ldots, x_{k,m}^{\bar{l}_k} \in \mathcal{B}(x_k, \gamma_\epsilon \epsilon_{k,\bar{l}_k}^\sigma)$. So, we admit the following hypotheses on problem (5):

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

$$\phi_j \left(x_{k,i_j}^{\bar{l}_k} \right) > \phi_s \left(x_{k,i_j}^{\bar{l}_k} \right), \ \forall s \in \{1, \ldots, p\}, \ s \neq j. \tag{10}$$

For an easier exposition of our ideas, we will write without loss of generality that

$$\phi_i \left(x_{k,i}^{\bar{l}_k} \right) > \phi_s \left(x_{k,i}^{\bar{l}_k} \right), \ \forall s \in \{1, \ldots, p\}, \ s \neq i \text{ and } i \in \mathcal{I}(x_*), \tag{11}$$

since by a simple rearrangement of the sampled points in (10) the inequality (11) holds;

H2) The first $r+1$ constraints are active at the solution;

H3) Only the first $r+1$ constraints are active at the solution[3].

Remark 2 Notice that supposing H3, we are implicitly asking that the trust-region constraint is not active, a common assumption made in smooth local convergence analysis.

Under those hypotheses, one can rewrite (5) as the following optimization problem

$$\begin{aligned} \min_{(d,z) \in \mathbb{R}^{n+1}} \quad & z + \frac{1}{2} d^T H_k d \\ \text{s.t.} \quad & \phi_i \left(x_{k,i}^{\bar{l}_k} \right) + \nabla \phi_i \left(x_{k,i}^{\bar{l}_k} \right)^T \left(x_k + d - x_{k,i}^{\bar{l}_k} \right) = z, \ \ 1 \leq i \leq r+1. \end{aligned} \tag{12}$$

[3] 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

$$\begin{aligned} \min_{d\in\mathbb{R}^n} \ & \phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) + \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \left(x_k + d - x_{k,r+1}^{\bar{l}_k}\right) + \frac{1}{2} d^T H_k d \\ \text{s.t.} \ & \tilde{\Phi}_k + \tilde{J}_k d = 0, \end{aligned} \tag{13}$$

where $\tilde{\Phi}_k \in \mathbb{R}^r$ with

$$\begin{aligned} (\tilde{\Phi}_k)_i := {} & \phi_i\left(x_{k,i}^{\bar{l}_k}\right) + \nabla\phi_i\left(x_{k,i}^{\bar{l}_k}\right)^T \left(x_k - x_{k,i}^{\bar{l}_k}\right) \\ & - \left[\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) + \nabla\phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right)^T \left(x_k - x_{k,r+1}^{\bar{l}_k}\right)\right], \ i \in \{1,\ldots,r\}, \end{aligned}$$

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 (5) can be viewed as a local approximation of

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & \phi_{r+1}(x) \\ \text{s.t.} \ & \Phi(x) = 0, \end{aligned} \tag{14}$$

where

$$\Phi(x) := \begin{pmatrix} \phi_1(x) - \phi_{r+1}(x) \\ \vdots \\ \phi_r(x) - \phi_{r+1}(x) \end{pmatrix}.$$

In fact, it is straightforward to see that x_* is also a local minimizer for (14). Hence, under special circumstances, (5) is equivalent to an optimization problem that has only functions in C^2.

To start our analysis, a new definition is necessary (a more general definition can be found in [26]).

Definition 4 (U,V-spaces) Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is the continuous objective function of problem (2) and x is any point in $\mathbb{R}^n$. 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 the smooth and nonsmooth subspaces of f at x, respectively.

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 tends to recover the smooth subspace of f at x_* when x approaches x_*. Therefore, we denote by J_x the Jacobian of $\Phi(x)$ and by $Z_x^{\triangleleft}$ 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).

In light of Remark 1, due to the Assumption 2, 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 $J_x^{\triangleleft} \in \mathbb{R}^{n \times r}$ such that $J_x J_x^{\triangleleft} = I_r$. Moreover, by [2, Lemma 14.3], one can see that there is only one map

$$\begin{aligned} Z : \mathbb{R}^n &\longrightarrow \mathbb{R}^{(n-r)\times n} \\ x &\longmapsto Z_x \end{aligned}$$

such that $Z_x J_x^{\triangleleft}$ is a null matrix, $Z_x Z_x^{\triangleleft} = I_{n-r}$ and the following relation holds

$$Z_x^{\triangleleft} Z_x + J_x^{\triangleleft} J_x = I_n. \tag{15}$$

So, we may divide $\mathbb{R}^n$ in two subspaces, generated by the columns of $Z_x^{\triangleleft}$ and $J_x^{\triangleleft}$, respectively.

Now, coming back to the optimization problem (14), we define its Lagrangian function $\mathcal{L}(x, \lambda) : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}$ as

$$\mathcal{L}(x, \lambda) = \phi_{r+1}(x) + \lambda^T \Phi(x). \tag{16}$$

Hence, Remark 1 yields that the feasible set of problem (14) 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

$$g(x) := Z_x^{\triangleleft T} \nabla_x \mathcal{L}(x, \lambda_*) = Z_x^{\triangleleft T} \nabla \phi_{r+1}(x). \tag{17}$$

Moreover, for not overloading the proofs that will follow, we also define

$$A_k := I_n - Z_{x_k}^{\triangleleft} \hat{H}_k^{-1} Z_{x_k}^{\triangleleft}{}^T H_k, \tag{18}$$

with

$$\hat{H}_k := Z_{x_k}^{\triangleleft}{}^T H_k Z_{x_k}^{\triangleleft}.$$

Below, we present a theorem that establishes the exact solution $d_{k,\bar{l}_k}$ obtained in (5) when it is equivalent to (13). For this result and the subsequent ones, we define

$$\tau_{k,\bar{l}_k} := \max_{1 \le i \le r+1} \left\| x_{k,i}^{\bar{l}_k} - x_k \right\|.$$

Theorem 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

$$d_{k,\bar{l}_k}^U := -Z_{x_k}^{\triangleleft} \hat{H}_k^{-1} g(x_k) + \rho_k^U \quad \textit{and} \quad d_{k,\bar{l}_k}^V := -A_k J_{x_k}^{\triangleleft} \Phi(x_k) + \rho_k^V,$$

with

$$\rho_k^U = -Z_{x_k}^{\lhd} \hat{H}_k^{-1} Z_{x_k}^{\lhd}{}^T \overline{\rho}_k \quad \textit{and} \quad \rho_k^V = -A_k J_{x_k}^{\lhd} \hat{\rho}_k,$$

for some $\overline{\rho}_k \in \mathbb{R}^n$ *and* $\hat{\rho}_k \in \mathbb{R}^r$ *satisfying*

$$\|\overline{\rho}_k\| = O\left(\tau_{k,\bar{l}_k}\right) \quad \textit{and} \quad \|\hat{\rho}_k\| = O\left(\tau_{k,\bar{l}_k}^2\right) + O\left(\tau_{k,\bar{l}_k}\right) O\left(\nu_k\right).$$

Proof First, we consider the Karush-Kuhn-Tucker conditions of problem (13), which tell us that the solution $d_{k,\bar{l}_k}$ must satisfy

$$\tilde{\Phi}_k + \tilde{J}_k d_{k,\bar{l}_k} = 0 \tag{19}$$

and

$$\nabla \phi_{r+1}\left(x_{k,r+1}^{\bar{l}_k}\right) + H_k d_{k,\bar{l}_k} + \tilde{J}_k^T \tilde{\lambda} = 0, \tag{20}$$

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 (19) and (20) and by

$$\|d_{k,\bar{l}_k}\|_\infty \leq \Delta_{k,\bar{l}_k} \leq \gamma_\Delta \nu_k,$$

that

$$\begin{aligned} \Phi(x_k) + J_{x_k} d_{k,\bar{l}_k} + \left[\tilde{\Phi}_k - \Phi(x_k)\right] + \left[\tilde{J}_k - J_{x_k}\right] d_{k,\bar{l}_k} &= 0 \\ \Phi(x_k) + J_{x_k} d_{k,\bar{l}_k} + \hat{\rho}_k &= 0 \end{aligned} \tag{21}$$

and

$$\nabla \phi_{r+1}\left(x_k\right) + H_k d_{k,\bar{l}_k} + J_{x_k}^T \tilde{\lambda} + \overline{\rho}_k = 0, \tag{22}$$

where $\|\hat{\rho}_k\| = O\left(\tau_{k,\bar{l}_k}^2\right) + O\left(\tau_{k,\bar{l}_k}\right) O\left(\nu_k\right)$ 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 (21), 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 (22) 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 ${Z^{\triangleleft}_{x_k}}^T H_k A_k = 0$, we complete the proof by noticing that

$$\alpha_U = -\hat{H}_k^{-1} g(x_k) - \hat{H}_k^{-1} {Z^{\triangleleft}_{x_k}}^T \overline{\rho}_k \Rightarrow d^U_{k,\bar{l}_k} = -Z^{\triangleleft}_{x_k} \hat{H}_k^{-1} g(x_k) + \rho^U_k,$$

where $\rho^U_k = -Z^{\triangleleft}_{x_k} \hat{H}_k^{-1} {Z^{\triangleleft}_{x_k}}^T \overline{\rho}_k$. □

With this theorem in hand, we are able to prove a simple corollary.

Corollary 1 *Under the assumptions of Theorem 2, we have that*

$$\|\Phi(x_{k+1})\| = O(\nu_k^2).$$

Proof Since $\phi_i \in C^2$, $\|d_{k,\bar{l}_k}\| \leq \nu_k$ and $\tau_{k,\bar{l}_k} \leq \nu_k$, it yields that

$$\begin{aligned}\|\Phi(x_{k+1})\| &\leq \|\Phi(x_k) + J_{x_k} d_{k,\bar{l}_k}\| + O(\nu_k^2)\\ &\leq \|\Phi(x_k) - \Phi(x_k)\| + \|\hat{\rho}_k\| + O(\nu_k^2)\\ &= O(\nu_k^2),\end{aligned}$$

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 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 3 *Suppose that $x_k \to x_*$, where $x_* \in \mathbb{R}^n$ is a local minimizer for f presented in (2). 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*

$$\begin{array}{rcl} Z^{\triangleleft} : \mathbb{R}^n & \longrightarrow & \mathbb{R}^{n\times(n-r)} \\ x & \longmapsto & Z^{\triangleleft}_x \end{array} \quad \textit{and} \quad \begin{array}{rcl} J^{\triangleleft} : \mathbb{R}^n & \longrightarrow & \mathbb{R}^{n\times r} \\ x & \longmapsto & J^{\triangleleft}_x \end{array}$$

are all Lipschitz continuous functions close to x_* and that the reduced gradient given in (17) satisfies $g \in C^1$. Moreover, assume that $H_k \to H_*$ is such that

$$H_* = \nabla^2_{xx}\mathcal{L}(x_*, \lambda_*) + \gamma J^T_{x_*} J_{x_*}, \quad \textit{for some } \gamma \geq 0.^4 \tag{23}$$

Additionally, suppose that close to x_* we have that $\|H_k - H_*\| = O(\|x_k - x_*\|)$. Then, the following relation holds

$$\|x_{k+1} - x_*\| = O(\|x_k - x_*\|^2) + \rho^U_k + \rho^V_k, \quad \textit{for } k \in \mathcal{K}.$$

[4] Notice that with this equation, we are, in an implicit way, saying that ${Z^{\triangleleft}_{x_*}}^T \nabla^2_{xx}\mathcal{L}(x_*, \lambda_*) Z^{\triangleleft}_{x_*}$ is a positive definite matrix (since the matrices H_k are all assumed to be positive definite and satisfy Assumption 1), and therefore, x_* must be an isolated local minimizer. Moreover, we only ask that, for all directions d in the null space of J_{x_*}, the effect of the matrices H_k upon d converge to $\nabla^2_{xx}\mathcal{L}(x_*, \lambda_*)d$ (an assumption close to the Dennis-Moré condition [2]).

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 (18), for x_k close enough to x_*, we have $\|A_k - A_*\| = O(\|x_k - x_*\|)$, where

$$A_* := I_n - Z^{\lhd}_{x_*} \hat{H}^{-1}_* Z^{\lhd}_{x_*}{}^T H_*, \text{ with } \hat{H}_* := Z^{\lhd}_{x_*}{}^T H_* Z^{\lhd}_{x_*}.$$

Using this fact, considering the Taylor expansion of the map Φ around x_* in the relation $(*)$ below and noticing that $J^{\lhd}$ is Lipschitz continuous and a bounded map around x_* in $(**)$, we have for a sufficiently small neighborhood of x_* that

$$\begin{aligned}
\tilde{x}_{k+1} - x_* &= x_k - x_* - A_k J^{\lhd}_{x_k} \Phi(x_k) + \rho^V_k \\
&\overset{(*)}{=} x_k - x_* - A_k J^{\lhd}_{x_k} J_{x_*}(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho^V_k \\
&= x_k - x_* - A_* J^{\lhd}_{x_*} J_{x_*}(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho^V_k \\
&\quad - \left[A_k \left(J^{\lhd}_{x_k} - J^{\lhd}_{x_*}\right) + (A_k - A_*) J^{\lhd}_{x_*}\right] J_{x_*}(x_k - x_*) \\
&\overset{(**)}{=} x_k - x_* - A_* J^{\lhd}_{x_*} J_{x_*}(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho^V_k .
\end{aligned}$$

Consequently, taking into account the relation (see [2, Section 14.5])

$$g'(x_*) = Z^{\lhd}_{x_*}{}^T \nabla^2_{xx} \mathcal{L}(x_*, \lambda_*)$$

in (●), the Lipschitz property around x_* of the maps $Z^{\lhd}$ and $\hat{H}^{-1}$ in (●●), the relation (23) in (▲) and the relation (15) in (▲▲), we have

$$\begin{aligned}
x_{k+1} - x_* &= \tilde{x}_{k+1} - x_* - Z^{\lhd}_{x_k} \hat{H}^{-1}_k g(x_k) + \rho^U_k \\
&\overset{(\bullet)}{=} \tilde{x}_{k+1} - x_* - Z^{\lhd}_{x_k} \hat{H}^{-1}_k Z^{\lhd}_{x_*}{}^T \nabla^2_{xx} \mathcal{L}(x_*, \lambda_*)(x_k - x_*) \\
&\quad + O(\|x_k - x_*\|^2) + \rho^U_k \\
&\overset{(\bullet\bullet)}{=} \tilde{x}_{k+1} - x_* - Z^{\lhd}_{x_*} \hat{H}^{-1}_* Z^{\lhd}_{x_*}{}^T \nabla^2_{xx} \mathcal{L}(x_*, \lambda_*)(x_k - x_*) \\
&\quad + O(\|x_k - x_*\|^2) + \rho^U_k \\
&\overset{(\blacktriangle)}{=} \tilde{x}_{k+1} - x_* - Z^{\lhd}_{x_*} \hat{H}^{-1}_* Z^{\lhd}_{x_*}{}^T H_*(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho^U_k \\
&= A_*(x_k - x_*) - A_* J^{\lhd}_{x_*} J_{x_*}(x_k - x_*) + O(\|x_k - x_*\|^2) \\
&\quad + \rho^U_k + \rho^V_k \\
&= A_*(I - J^{\lhd}_{x_*} J_{x_*})(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho^U_k + \rho^V_k \\
&\overset{(\blacktriangle\blacktriangle)}{=} A_* Z^{\lhd}_{x_*} Z_{x_*}(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho^U_k + \rho^V_k .
\end{aligned}$$

Hence, since $A_* Z^{\lhd}_{x_*} = 0$, it yields that

$$\|x_{k+1} - x_*\| = O(\|x_k - x_*\|^2) + \rho^U_k + \rho^V_k ,$$

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: $\epsilon^{\sigma}_{k,\bar{l}_k} = o(\|x_k - x_*\|)$. 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 $\epsilon^{\sigma}_{k,\bar{l}_k}$ is impossible, since this demands to know x_*. However, taking a careful look at the Karush-Kuhn-Tucker conditions of (5), we have that

$$d_{k,\bar{l}_k} = -H_k^{-1}\left(G_{k,\bar{l}_k}\lambda_{k,\bar{l}_k} + \omega_{k,\bar{l}_k}\right). \tag{24}$$

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 $\mathcal{B}(x_k, \gamma_\epsilon \epsilon^{\sigma}_{k,\bar{l}_k})$, it is not absurd to expect that, for an appropriate value of $\sigma \in [1, 2]$, the following will hold for a reasonable amount of times

$$\tau_{k,\bar{l}_k} = o(\|x_k - x_*\|). \tag{25}$$

One can argue that it would be better to let σ be greater than two in order to have a smaller sampling radius, and consequently, to increase the chance of (25) 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 3 We stress that $\epsilon^{\sigma}_{k,\bar{l}_k} = o(\|x_k - x_*\|)$ is a desirable result, but by no means, it is a necessary condition to a rapid movement towards the solution of the optimization problem. Let us consider that $\epsilon^{\sigma}_{k,\bar{l}_k}$ 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 evinces the importance of Corollary 1). As a result, a good approximation of the value $\|x_k - x_*\|$ just allows the method to increase the probability of (25) to happen. Finally, we also highlight that for the case that $V = \mathbb{R}^n$, the equality (25) can be traded for $\tau_{k,\bar{l}_k} = O(\|x_k - x_*\|)$ without affecting the superlinear convergence (to see this, just notice the properties of ρ^V_k).

Remark 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 where 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.,

$d_{k,\bar{l}_k} = d^U_{k,\bar{l}_k}$ with $Z_x^{\triangleleft} = I_n$ for all x around x_*. Now, considering $r = n$, we see that the method is approaching a point where 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., $d_{k,\bar{l}_k} = d^V_{k,\bar{l}_k}$ 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 $r = n$).

5 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 had 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.

All the problems were solved 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 `active-set` as the algorithmic choice and 10^{-12} as the tolerances `TolX` and `TolFun` and 10^{-8} (default value) as `TolCon`. Moreover, for all functions we have chosen random starting points such that $\|x_0\|_\infty \leq 2$ 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 [5] but with a nonnormalized search direction (a variant introduced by Kiwiel [22], 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)[5]. For completeness, we present the parameter values used in Algorithm 1: $m = 2n$; $\nu_0 = 10^{-6}$; $\epsilon_0 = 10^{-1}$; $\nu_{\text{opt}} = 10^{-6}$; $\epsilon_{\text{opt}} = 10^{-6}$; $\theta_\nu = 1$; $\theta_\epsilon = 10^{-1}$; $\gamma = 0.5$; $\beta = 0$; $\alpha_k = 1$ and $H_k = I$.

The implementation of GraFuS is also available[6] and we have used it to produce the numerical results. 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

[5] The GS code can be found at http://cs.nyu.edu/overton/papers/gradsamp/alg/.

[6] The authors freely provide the GraFuS code:

GraFuS code Test function Script

$\theta = 0.5$. The value of σ in Step 1 was set as $\sigma = 1 + 0.5^{(l+2)/4} \mod (l+1, 2)$. More elaborated ways to set σ were considered but none of them presented to be consistently better than this procedure. It would be desirable to have a way to identify the relevance of the U component in the search direction of each iteration to have a tunned σ, however this requires a recognition of the U and V spaces at the solution x_*. An attempt to identify those spaces 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 did not obtain a satisfactory recognition.

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 (23). 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 [14] for the optimization problem that appears in (14).

5.1 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 (16). 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 (16) and λ_+ and λ_- are vectors that try to approximate the multiplier λ_* that fulfills (17). 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

$$\left\|G_{k_+,\bar{l}_{k_+}} \lambda_{k_+,\bar{l}_{k_+}}\right\|_\infty \leq 10\nu_{k_+} \quad \text{and} \quad \left\|G_{k_-,\bar{l}_{k_-}} \lambda_{k_-,\bar{l}_{k_-}}\right\|_\infty \leq 10\nu_{k_-}.$$

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 $q^T p < 0.2 p^T H p$ then compute a new vector q by Powell's correction (see [2, Subsection 18.2]).

Step 2. Update H:

$$H \leftarrow H - \frac{H p p^T H}{p^T H p} + \frac{q q^T}{q^T p}.$$

Step 3. Use the subsequent matrices H_k as H until the GraFuS algorithm finds another iteration $\hat{k}$ such that

$$\left\| G_{\hat{k},\bar{l}_{\hat{k}}} \lambda_{\hat{k},\bar{l}_{\hat{k}}} \right\|_\infty \leq 10\nu_{\hat{k}}.$$

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 [27] can be performed (although, in such a case, we have to assume that for all iterates the function f will be differentiable and that Assumption 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 have 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 $\Diamond$ 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 presented to be closer to the solution, it 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.

5.2 Test functions with $V = \mathbb{R}^n$

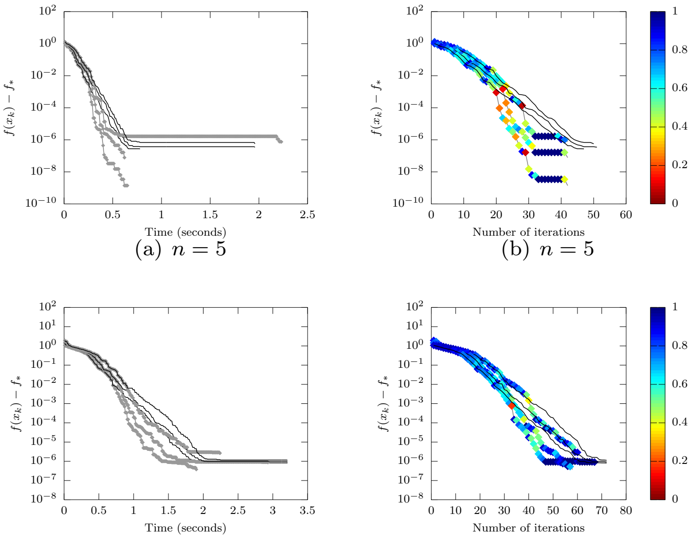
We present two nonconvex and nonsmooth functions [17] that, at the solution point, have the whole space $\mathbb{R}^n$ as the V space:

F1) Active faces (defined for all number of variables n)

$$f(x) = \max\left\{ g\left(-\sum_{i=1}^{n} x_i \right), g(x_i) \right\}, \quad \text{with} \quad g(z) = \log(|z| + 1);$$

F2) Chained Mifflin 2 (defined for all number of variables $n \geq 2$)

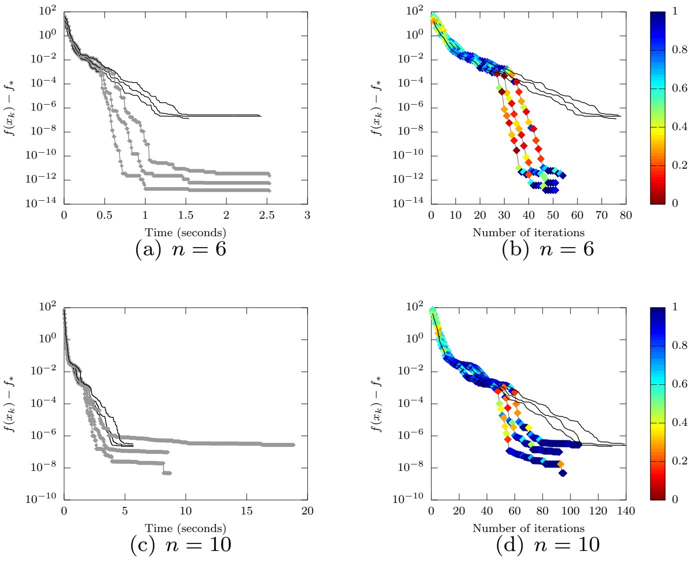
$$f(x) = \sum_{i=1}^{n-1} \left(-x_i + 2(x_i^2 + x_{i+1}^2 - 1) + 1.75|x_i^2 + x_{i+1}^2 - 1| \right).$$



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

(c) $n = 10$ (d) $n = 10$

Fig. 1 Medians and quartiles of twenty runs of GS and GraFuS methods for function **F1**. For both number of variables we have $x_* = 0$.



(a) $n = 6$ (b) $n = 6$

(c) $n = 10$ (d) $n = 10$

Fig. 2 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 $x_* \approx (0.8152, 0.5792, 0.7362, 0.6767, 0.7362, 0.6767, 0.7362, 0.6767, 0.7362, 0)^T$.

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 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 2 must not hold for the functions **F1** and **F2**. Fortunately, this fact does not prevent GraFuS to converge for both functions (see Figures 1 and 2).

The good behavior in the absence of the validity of Assumption 2 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 1 (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 best 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.

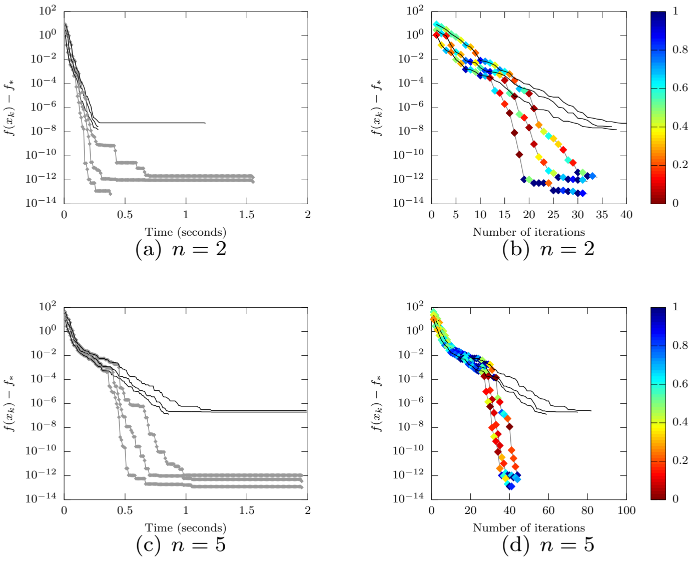
Remark 5 The functions inside the subsection of maps with multiple stationary points do also satisfy $V = \mathbb{R}^n$. However, we have chosen to separate them from **F1** and **F2** because they have an additional property.

5.3 Test functions with $V \neq \mathbb{R}^n$

In the previous subsection we only presented functions for which the U space is empty 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 [17,38]: **F2** presented previously (but now with $n \in \{2, 5\}$) and

F3) Generalized Rosenbrock function (defined for all number of variables $n \geq 2$)

$$f(x) = \sum_{i=1}^{n-1} \left(\frac{10i}{n} \left| x_{i+1} - \frac{i}{n} x_i^2 \right| + \frac{i}{n}(1 - x_i)^2 \right).$$



(a) $n = 2$ (b) $n = 2$

(c) $n = 5$ (d) $n = 5$

Fig. 3 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 $x_* \approx (0.8152, 0.5792, 0.7071, 0.7071, 0)^T$.

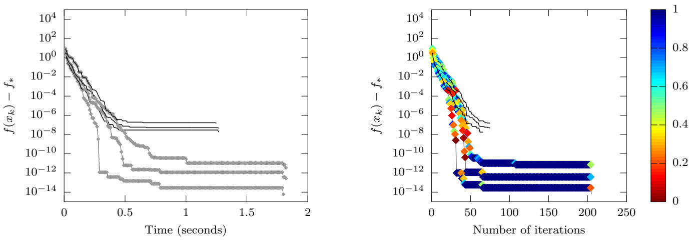
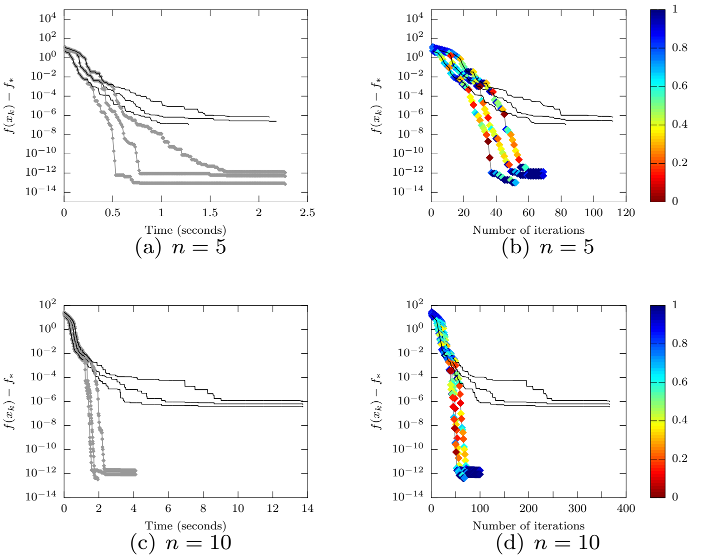


Fig. 4 Medians and quartiles of twenty runs of GS and GraFuS methods for function **F3** (with $n = 2$). We have $x_* = (1, 0.5)^T$.

F4) Chained crescent I (defined for all number of variables $n \geq 2$)

$$f(x) = \max\left\{ \sum_{i=1}^{n-1} \left(x_i^2 + (x_{i+1}-1)^2 + x_{i+1} - 1\right), \right.$$
$$\left. \sum_{i=1}^{n-1} \left(-x_i^2 - (x_{i+1}-1)^2 + x_{i+1} + 1\right) \right\}.$$

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



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

(c) $n = 10$ (d) $n = 10$

Fig. 5 Medians and quartiles of twenty runs of GS and GraFuS methods for function **F4**. For both number of variables we have $x_* = 0$.

instances with $n = 5$ and $n = 10$. The results can be seen in Figures 3 - 5 and the rapid convergence behavior is also observed in some iterations of GraFuS.

5.4 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 [17,38]:

F5) Chained crescent II (defined for all number of variables $n \geq 2$)

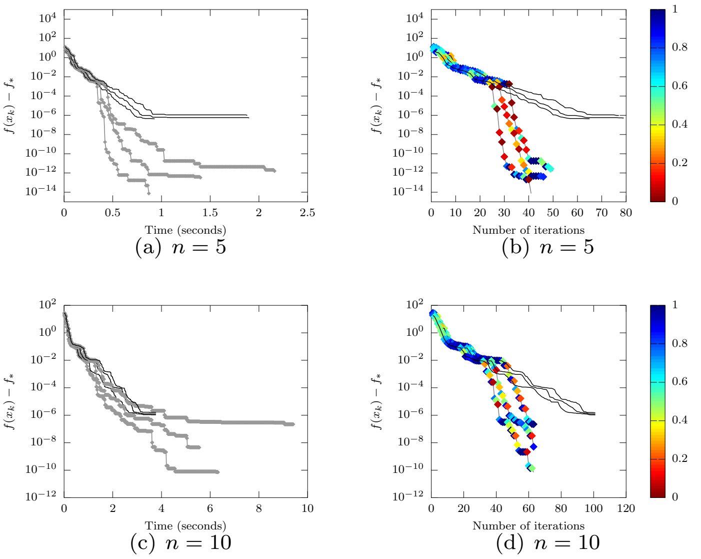
$$f(x) = \sum_{i=1}^{n-1} \max\left\{\left(x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1\right),\right.$$
$$\left.\left(-x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1\right)\right\};$$

F6) Problem 17 of Test 29 of [38] (defined for all n multiple of 5)

$$f(x) = \max_{1 \leq i \leq n} \left\{ \left| 5 - (j+1)(1 - \cos x_i) - \sin x_i - \sum_{k=5j+1}^{5j+5} \cos x_k \right| \right\},$$

with $j = \lfloor (i - 1/5) \rfloor$.

The results can be found in Figures 6 and 7. Again, it is possible to find iterates for which the algorithm moves fast to the solution, enlarging the results previously obtained.



(a) $n = 5$

(b) $n = 5$

(c) $n = 10$

(d) $n = 10$

Fig. 6 Medians and quartiles of twenty runs of GS and GraFuS methods for function **F5**. For both number of variables we have $x_* = 0$.

5.5 Test functions without an appropriate maximum representation

The next functions can be seen in [17,38] and they cannot be written as the maximum of sufficiently smooth functions:

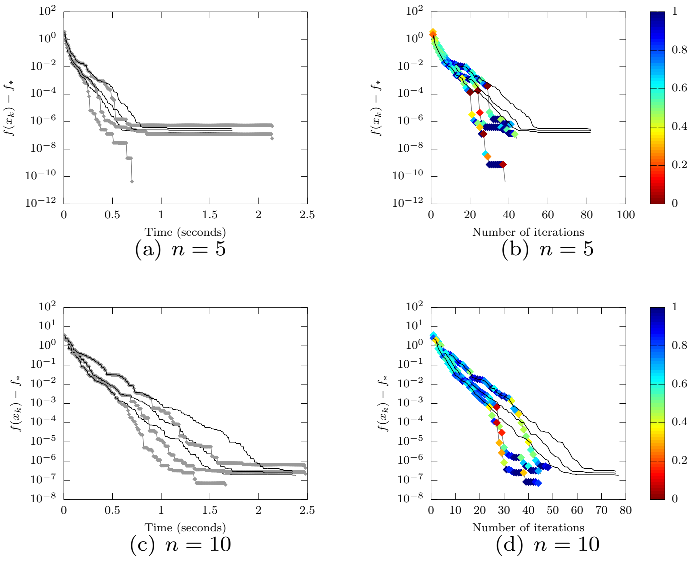
F7) Nonsmooth generalization of Brown function 2 (defined for all number of variables $n \geq 2$)

$$f(x) = \sum_{i=1}^{n-1} \left(|x_i|^{x_{i+1}^2+1} + |x_{i+1}|^{x_i^2+1} \right);$$

F8) Nonsmooth and nonconvex toy problem (defined for all number of variables $n \geq 2$)

$$f(x) = \sqrt{g(x)}, \quad \text{with} \quad g(x) = \delta + \sqrt{x^T A x} + x^T B x,$$

where $\delta \in (0, 1)$ is a fixed parameter, $A = \text{diag}(1, 0, 1, 0, \ldots)$ and $B = \text{diag}(1, 1/4, \ldots, 1/n^2)$.



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

(c) $n = 10$ (d) $n = 10$

Fig. 7 Medians and quartiles of twenty runs of GS and GraFuS methods for function **F6**. For both number of variables we have $x_* = 0$.

For the first 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 $a \geq 0$. 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 the function **F8**, which primarily appeared in a preprint of [27]. 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

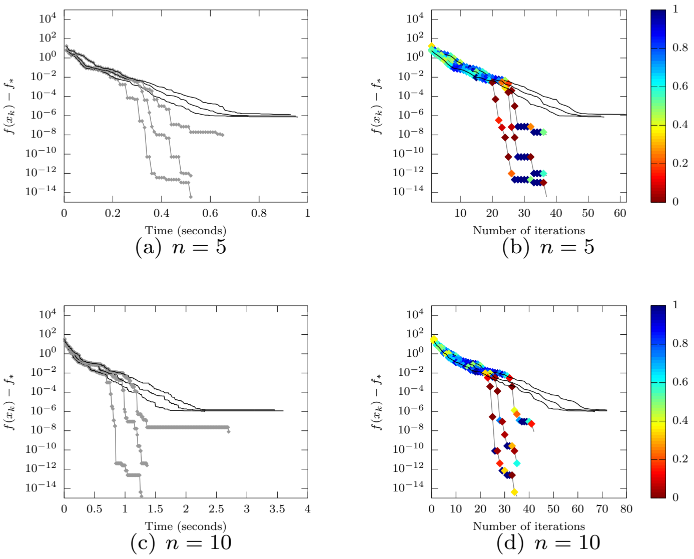
$$\nabla g(x) = (x^T Ax)^{-1/2}Ax + 2Bx$$

and

$$\nabla^2 g(x) = -(x^T Ax)^{-3/2}Ax(Ax)^T + (x^T Ax)^{-1/2}A + 2B.$$

Consequently, if one could have a maximum representation of f as in (2), 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 8 and 9), we see that this fact is not an obstacle for GraFuS to present a rapid convergence behavior for both functions.



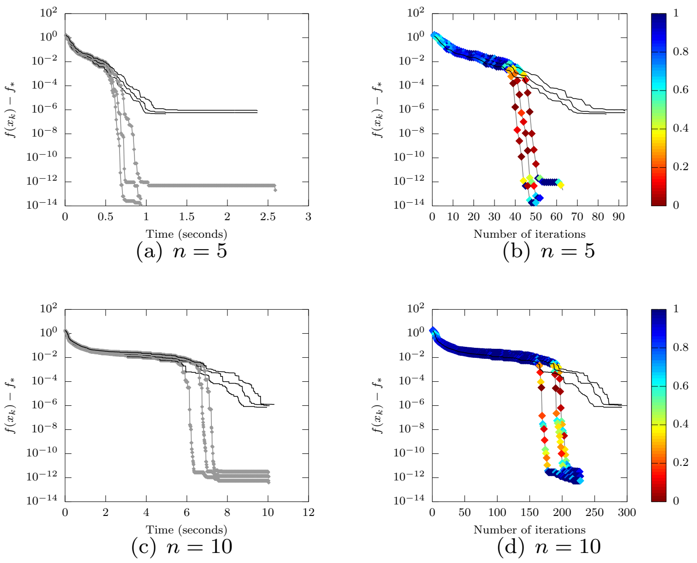
(a) $n = 5$ (b) $n = 5$ (c) $n = 10$ (d) $n = 10$

Fig. 8 Medians and quartiles of twenty runs of GS and GraFuS methods for function **F7**. For both number of variables we have $x_* = 0$.

6 Final remarks and conclusion

This manuscript 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, its rapid behavior might



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

(c) $n = 10$ (d) $n = 10$

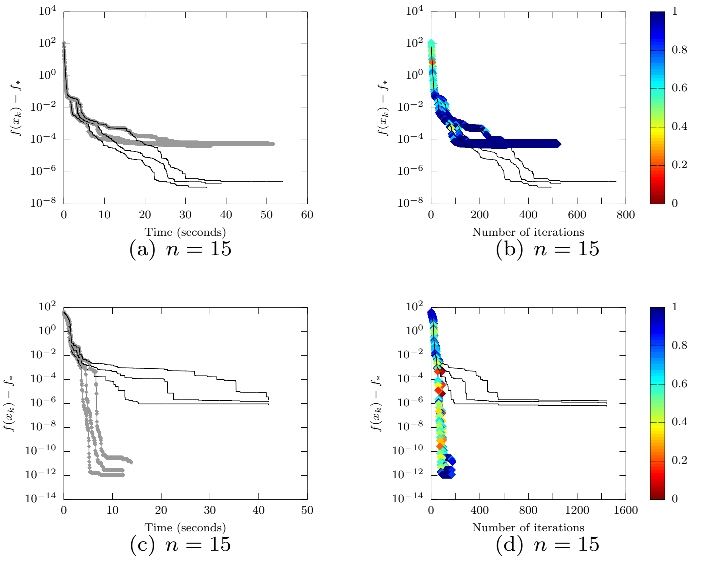
Fig. 9 Medians and quartiles of twenty runs of GS and GraFuS methods for function **F8** with $\delta = 10^{-2}$. For both number of variables we have $x_* = 0$.

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 10). Nevertheless, if we allow the GS method to work a few more iterations, reasonable results are recovered for the function **F2** (see Figure 11).

The matters of efficiency and applicability of the method are not treated properly in this manuscript, 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.

Finally, we end these final remarks with two questions that naturally arise from some of the numerical results obtained in the previous section:

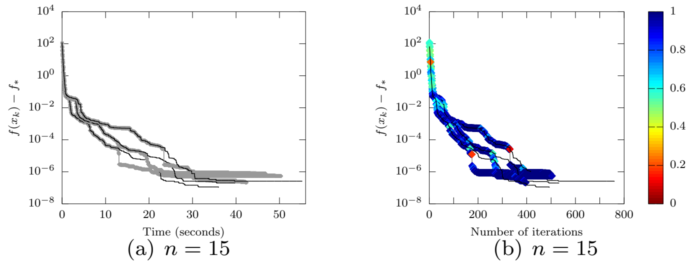
- under which conditions could we establish $\|H_k - H_*\| = O(\|x_k - x_*\|)$ in Theorem 3?
- would it be possible to have convergence results with more general assumptions?



(a) $n = 15$ (b) $n = 15$

(c) $n = 15$ (d) $n = 15$

Fig. 10 Medians and quartiles of twenty runs of GS and GraFuS methods for functions **F3** (top) and **F5** (bottom).



(a) $n = 15$ (b) $n = 15$

Fig. 11 Medians and quartiles of twenty runs of GS and GraFuS methods for function **F3**, but allowing the GS method to work a few more iterations before we start GraFuS.

References

1. Balinski, M.L., Wolfe, P.: Nondifferentiable Optimization, vol. 3. Math. Programming Studies., USA (1975)
2. Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.A.: Numerical optimization: theoretical and practical aspects, 2nd edn. Springer-Verlag Berlin Heidelberg (2006)
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
4. Burke, J.V., Lewis, A.S., Overton, M.L.: Approximating subdifferentials by random sampling of gradients. Mathematics of Operations Research **27**(3), 567–584 (2002)
5. Burke, J.V., Lewis, A.S., Overton, M.L.: A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM Journal on Optimization **15**(3), 751–779 (2005)

6. Clarke, F.H.: Optimization and nonsmooth analysis, vol. 5. SIAM, Montreal, Canada (1990)
7. Clarke, F.H., Ledyaev, Y.S., Stern, R.J., Wolenski, P.R.: Nonsmooth analysis and control theory, vol. 178. Springer, New York (2008)
8. Curtis, F.E., Overton, M.L.: A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization. SIAM Journal on Optimization **22**(2), 474–500 (2012)
9. Curtis, F.E., Que, X.: An adaptive gradient sampling algorithm for non-smooth optimization. Optimization Methods and Software **28**(6), 1302–1324 (2013)
10. Do, T.M.T., Artières, T.: Regularized bundle methods for convex and non-convex risks. The Journal of Machine Learning Research **13**(1), 3539–3583 (2012)
11. Dotta, D., Silva, A.S., Decker, I.C.: Design of power system controllers by nonsmooth, nonconvex optimization. In: Power Energy Society General Meeting, 2009. PES '09. IEEE, pp. 1–7 (2009)
12. Du, D.Z., Pardalos, P.M.: Minimax and applications, vol. 4. Springer US (2013)
13. Fuduli, A., Gaudioso, M., Giallombardo, G.: A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. Optimization Methods and Software **19**(1), 89–102 (2004)
14. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM Review **47**(1), 99–131 (2005)
15. Goldstein, A.A.: Optimization of Lipschitz continuous functions. Mathematical Programming **13**(1), 14–22 (1977)
16. Grothey, A., McKinnon, K.: A superlinearly convergent trust region bundle method. Report, Department of Mathematics & Statistics, Edinburgh University (1998)
17. Haarala, M., Miettinen, K., Mäkelä, M.M.: New limited memory bundle method for large-scale nonsmooth optimization. Optimization Methods and Software **19**(6), 673–692 (2004)
18. Helou, E.S., Santos, S.A., Simões, L.E.A.: On the differentiability check in gradient sampling methods. Optimization Methods and Software, Online (2016). DOI 10.1080/10556788.2016.1178262
19. Huber, G.: Gamma function derivation of n-sphere volumes. The American Mathematical Monthly **89**(5), 301–302 (1982)
20. Kelley Jr, J.E.: The cutting-plane method for solving convex programs. Journal of the Society for Industrial and Applied Mathematics **8**(4), 703–712 (1960)
21. Kiwiel, K.C.: Methods of descent for nondifferentiable optimization, vol. 1133. Springer Berlin Heidelberg (1985)
22. Kiwiel, K.C.: Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization. SIAM Journal on Optimization **18**(2), 379–388 (2007)
23. Lemaréchal, C., Mifflin, R.: Global and superlinear convergence of an algorithm for one-dimensional minimization of convex functions. Mathematical Programming **24**(1), 241–256 (1982)
24. Lemaréchal, C., Oustry, F., Sagastizábal, C.: The U-Lagrangian of a convex function. Transactions of the American Mathematical Society **352**(2), 711–729 (2000)
25. Lemaréchal, C., Sagastizábal, C.: Practical aspects of the moreau–yosida regularization: Theoretical preliminaries. SIAM Journal on Optimization **7**(2), 367–385 (1997)
26. Lewis, A.S.: Active sets, nonsmoothness, and sensitivity. SIAM Journal on Optimization **13**(3), 702–725 (2002)
27. Lewis, A.S., Overton, M.L.: Nonsmooth optimization via quasi-Newton methods. Mathematical Programming **141**(1-2), 135–163 (2013)
28. Lukšan, L., Vlček, J.: A bundle-Newton method for nonsmooth unconstrained minimization. Mathematical Programming **83**(1-3), 373–391 (1998)
29. Mäkelä, M.: Survey of bundle methods for nonsmooth optimization. Optimization Methods and Software **17**(1), 1–29 (2002)
30. Maréchal, P., Ye, J.J.: Optimizing condition numbers. SIAM Journal on Optimization **20**(2), 935–947 (2009)
31. Mifflin, R., Sagastizábal, C.: VU-decomposition derivatives for convex max-functions. In: M. Théra, R. Tichatschke (eds.) Ill-posed Variational Problems and Regularization Techniques, *Lecture Notes in Economics and Mathematical Systems*, vol. 477, pp. 167–186. Springer Berlin Heidelberg (1999)

32. Mifflin, R., Sagastizábal, C.: A VU-algorithm for convex minimization. Mathematical Programming **104**(2-3), 583–608 (2005)
33. Mifflin, R., Sagastizábal, C.: A science fiction story in nonsmooth optimization originating at IIASA. In: M. Grötschel (ed.) Documenta Mathematica Optimization Stories, pp. 291–300. Deutschen Mathematiker-Vereinigung, Bielefeld (2012)
34. Moreau, J.J., Panagiotopoulos, P.D.: Nonsmooth mechanics and applications, vol. 302. Springer, Vienna (2014)
35. Oliveira, W., Sagastizábal, C.: Bundle methods in the XXIst century: A bird's-eye view. Pesquisa Operacional **34**(3), 647–670 (2014)
36. Outrata, J., Kočvara, M., Zowe, J.: Nonsmooth approach to optimization problems with equilibrium constraints: theory, applications and numerical results, vol. 28. Kluwer Academic Publishers, The Netherlands (2013)
37. Peng, C., Jin, X., Shi, M.: Epidemic threshold and immunization on generalized networks. Physica A: Statistical Mechanics and its Applications **389**(3), 549–560 (2010)
38. Skajaa, A.: Limited memory BFGS for nonsmooth optimization. Master's thesis, Courant Institute of Mathematical Science, New York University (2010)
39. Wang, F.C., Chen, H.T.: 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)
40. Zhang, J., Kim, N.H., Lasdon, L.: An improved successive linear programming algorithm. Management Science **31**(10), 1312–1331 (1985)

7 Appendix

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 on the fact that we are assuming the irrelevance of the trust-region constraint (which is a common assumption on smooth convergence analysis), 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 (13), we see that any additional active constraint will generate an additional active constraint to (13) in a way that it will be a linear combination of the first $n+1$ active constraints (by Remark 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 (5) is equivalent to minimize

$$\begin{aligned} \min_{(d,z)\in\mathbb{R}^{n+1}} \quad & z + \frac{1}{2}d^T H_k d \\ \text{s.t.} \quad & f\left(x_{k,i}^{\bar{l}_k}\right) + \nabla f\left(x_{k,i}^{\bar{l}_k}\right)^T \left(x_k + d - x_{k,i}^{\bar{l}_k}\right) = z, \ \ 1 \le i \le r+2, \end{aligned}$$

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

$$\begin{aligned} \min_{(d,z)\in\mathbb{R}^{n+1}} \quad & z + \frac{1}{2}d^T H_k d \\ \text{s.t.} \quad & \phi_i\left(x_{k,i}^{\bar{l}_k}\right) + \nabla \phi_i\left(x_{k,i}^{\bar{l}_k}\right)^T \left(x_k + d - x_{k,i}^{\bar{l}_k}\right) = z, \ \ 1 \le i \le r+1 \\ & \phi_j\left(x_{k,r+2}^{\bar{l}_k}\right) + \nabla \phi_j\left(x_{k,r+2}^{\bar{l}_k}\right)^T \left(x_k + d - x_{k,r+2}^{\bar{l}_k}\right) = z, \end{aligned}$$

whose dual optimization problem is written as

$$\begin{aligned} \max_{\lambda\in\mathbb{R}^{r+2}} \quad & \sum_{i=1}^{r+1} \lambda_i \left[\phi_i\left(x_{k,i}^{\bar{l}_k}\right) + \nabla \phi_i\left(x_{k,i}^{\bar{l}_k}\right)^T \left(x_k - x_{k,i}^{\bar{l}_k}\right)\right] \\ & + \lambda_{r+2}\left[\phi_j\left(x_{k,r+2}^{\bar{l}_k}\right) + \nabla \phi_j\left(x_{k,r+2}^{\bar{l}_k}\right)^T \left(x_k - x_{k,r+2}^{\bar{l}_k}\right)\right] \\ & - \frac{1}{2}\left\| \sum_{i=1}^{r+1} \lambda_i \nabla \phi_i(x_{k,i}^{\bar{l}_k}) + \lambda_{r+2} \nabla \phi_j(x_{k,r+2}^{\bar{l}_k}) \right\|_{H_k^{-1}}^2 \\ \text{s.t.} \quad & e^T \lambda = 1. \end{aligned} \tag{26}$$

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

$$\begin{aligned}\max_{\lambda \in \mathbb{R}^{r+1}} \quad & \sum_{i=1}^{r+1} \lambda_i \left[\phi_i \left(x_{k,i}^{\bar{l}_k} \right) + \nabla \phi_i \left(x_{k,i}^{\bar{l}_k} \right)^T \left(x_k - x_{k,i}^{\bar{l}_k} \right) - \phi_j \left(x_{k,r+2}^{\bar{l}_k} \right) \right. \\ & \left. - \nabla \phi_j \left(x_{k,r+2}^{\bar{l}_k} \right)^T \left(x_k - x_{k,r+2}^{\bar{l}_k} \right) \right] + \phi_j \left(x_{k,r+2}^{\bar{l}_k} \right) \\ & + \nabla \phi_j \left(x_{k,r+2}^{\bar{l}_k} \right)^T \left(x_k - x_{k,r+2}^{\bar{l}_k} \right) \\ & - \frac{1}{2} \left\| \sum_{i=1}^{r+1} \lambda_i \left[\nabla \phi_i (x_{k,i}^{\bar{l}_k}) - \nabla \phi_j (x_{k,r+2}^{\bar{l}_k}) \right] + \nabla \phi_j (x_{k,r+2}^{\bar{l}_k}) \right\|_{H_k^{-1}}^2 .\end{aligned}$$

Since the above problem is convex, its solution $\overline{\lambda} \in \mathbb{R}^{r+1}$ can be obtained by equalling 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

$$\begin{aligned}& \begin{pmatrix} \nabla \phi_1 \left(x_{k,1}^{\bar{l}_k} \right)^T - \nabla \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right)^T \\ \vdots \\ \nabla \phi_{r+1} \left(x_{k,r+1}^{\bar{l}_k} \right)^T - \nabla \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right)^T \end{pmatrix} H_k^{-1} \begin{pmatrix} \nabla \phi_1 \left(x_{k,1}^{\bar{l}_k} \right)^T - \nabla \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right)^T \\ \vdots \\ \nabla \phi_{r+1} \left(x_{k,r+1}^{\bar{l}_k} \right)^T - \nabla \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right)^T \end{pmatrix}^T \overline{\lambda} = \\ & \begin{pmatrix} \phi_1 \left(x_{k,1}^{\bar{l}_k} \right) + \nabla \phi_1 \left(x_{k,1}^{\bar{l}_k} \right)^T \left(x_k - x_{k,1}^{\bar{l}_k} \right) \\ \vdots \\ \phi_{r+1} \left(x_{k,r+1}^{\bar{l}_k} \right) + \nabla \phi_{r+1} \left(x_{k,r+1}^{\bar{l}_k} \right)^T \left(x_k - x_{k,r+1}^{\bar{l}_k} \right) \end{pmatrix} \\ & - \begin{pmatrix} \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right) + \nabla \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right)^T \left(x_k - x_{k,r+2}^{\bar{l}_k} \right) \\ \vdots \\ \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right) + \nabla \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right)^T \left(x_k - x_{k,r+2}^{\bar{l}_k} \right) \end{pmatrix} \\ & - \begin{pmatrix} \nabla \phi_1 \left(x_{k,1}^{\bar{l}_k} \right)^T - \nabla \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right)^T \\ \vdots \\ \nabla \phi_{r+1} \left(x_{k,r+1}^{\bar{l}_k} \right)^T - \nabla \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right)^T \end{pmatrix} H_k^{-1} \nabla \phi_{r+1} \left(x_{k,r+2}^{\bar{l}_k} \right) .\end{aligned}$$

Now, changing the points $x_{k,r+2}^{\bar{l}_k}$ for $x_{k,r+1}^{\bar{l}_k}$ and redefining

$$\tau_{k,\bar{l}_k} := \max_{1 \leq i \leq r+2} \left\| x_{k,i}^{\bar{l}_k} - x_k \right\| ,$$

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 (12) must satisfy the last linear system obtained above (not considering the remaining error vector) and, moreover,

$$\hat{\lambda}_{r+1} = 1 - \sum_{i=1}^{r} \hat{\lambda}_i. \tag{27}$$

Therefore, considering $\lambda^* \in \mathbb{R}^{r+2}$ the solution of (26) and using equation (27), 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 (24), 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 (12) 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 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.