A fast gradient and function sampling method for finite-max functions

Elias S. Helou1 · Sandra A. Santos2 · Lucas E. A. Simões2

Received: 25 September 2017 / Published online: 23 August 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract
This paper proposes an algorithm for the unconstrained minimization of a class of non-smooth and nonconvex functions that can be written as finite-max functions. A gradient and function-based sampling method is proposed which, under special circumstances, either moves superlinearly to a minimizer of the problem of interest or improves the optimality certificate. Global and local convergence analysis are presented, as well as examples that illustrate the obtained theoretical results.

Keywords Nonsmooth nonconvex optimization · Gradient sampling · Local superlinear convergence · Global convergence · Unconstrained minimization

1 Introduction

Problems involving continuous nonsmooth functions arise in many fields of science [41,50,51], playing a primary or a secondary role (e.g. subproblems) in different areas. A wide class of problems needs to cope with one or more minimizations of convex nonsmooth functions [46,49], which has been successfully solved by well established

Elias S. 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 302915/2016-8, FAPESP Grants 2013/05475-7 and 2013/07375-0 and PRONEX Optimization. Lucas E. A. Simões was supported by FAPESP Grants 2013/14615-7 and 2016/22989-2.

Lucas E. A. Simões
simoes.lea@gmail.com
Elias S. Helou
elias@icmc.usp.br
Sandra A. Santos
sandra@ime.unicamp.br

1 Institute of Mathematical Sciences and Computation, University of São Paulo, São Carlos SP, Brazil
2 Department of Applied Mathematics, University of Campinas, Campinas, SP, Brazil
optimization algorithms known as Bundle Methods [1,2,29,39]. However, a significant amount of problems involve minimizations of nonsmooth functions that are also nonconvex [14,15], which requires variations on the standard bundle ideas [18,30].

Recently, an algorithm known as Gradient Sampling (GS) [5,31] has gained attention for providing good alternatives to the difficulties that the Bundle Methods need to deal with if the function is not convex (see [39,48] 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 (i.e., when the search direction is not normalized in each iteration) to solve a smooth optimization problem, it tends to recover the steepest descent search direction as the sampling radius goes to zero at a fixed iterate. 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 at some 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 information. As a counterpart, by evaluating the gradients at 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 [25]. Therefore, this leads to a natural question: would it be possible to have a GS algorithm that can be understood as a generalization of Newton’s (or quasi-Newton) method for nonsmooth functions, meaning that it would locally converge faster than linearly?

This 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 [9–11], 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 (a concept that will be precisely defined in Lemma 6) and, for a special class of nonsmooth functions, the method will move superlinearly in some sense. This study is focused entirely on minimax problems, one of the foundations for the development of the nondifferentiable optimization field (see the introduction of [13]).

One might view our method as a GS algorithm that incorporates some elements of Bundle Methods developed over the years [22,38], but still keeps the GS facilities to handle nonconvex functions. This last characteristic is in agreement with Kiwiel’s expectation [31].

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

In order to prove a rapid local convergence result, the theory developed in this manuscript is based on the VU-decomposition of the space [35,42]. However, the method does not need to compute estimates of bases for such spaces. Roughly speaking, we show that our trust-region algorithm emulates the quasi-Newton techniques
into the $U$-space (a subspace where the objective function is locally smooth), whereas it combines effective cutting-plane features [17,28] into the $V$-space (the orthogonal complement of the $U$-space). 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 the computational effort of evaluating the gradient is sometimes proportional to the cost of evaluating the function value [21].

As a consequence of our attempt to move superlinearly to the solution of the optimization problem, the iterations of the proposed algorithm are more expensive when compared to the GS method. Therefore, although the global convergence of our algorithm is proven, the method should be viewed as an acceleration strategy for the GS algorithm to obtain a faster local convergence rate. Consequently, we propose that a potential user should use the GS method in the first iterations and switch to our algorithm in the final iterations.

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 [32–34,43]). The pursuit for such an algorithm has raised many researchers’ efforts (an enlightening review can be found in [44]) and up to our knowledge there is no method in the literature that rapidly converges when applied to the objective functions considered in this study. 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 global and local 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:

- $\text{co } \mathcal{X}$ is the convex hull of $\mathcal{X}$;
- $\text{cl } \mathcal{X}$ is the closure of $\mathcal{X}$;
- $\text{ri } \mathcal{X}$ is the relative interior of $\mathcal{X}$;
- $\text{int } \mathcal{X}$ is the interior of $\mathcal{X}$;
- $|\mathcal{X}|$ is the cardinality of $\mathcal{X}$;
- $B(x, r)$ is the Euclidean closed ball with center at $x$ and radius $r$;
- $\| \cdot \|$ is the Euclidean 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;
- $\mathcal{P}[x \in \mathcal{X}]$ 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

\[
\min_{x \in \mathbb{R}^n} f(x),
\]
where $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz function, continuously differentiable in an open subset with full measure $D \subset \mathbb{R}^n$. The function $f$ is not necessarily convex.

**Algorithm 1:** Gradient Sampling method.

**Step 0.** Given $x_0 \in D$, $m \in \mathbb{N}$ with $m \geq n + 1$, fixed real numbers $0 \leq \nu_0 < 0$, $0 \leq \epsilon_0 < \epsilon_0$ and $0 < \theta_\epsilon, \theta_\nu, \gamma, \beta < 1$, set $k = 0$.

**Step 1.** Choose $\{x_k, 1, \ldots, x_k, m\} \subset B(x_k, \epsilon_k)$ with randomly, independently and uniformly sampled elements. If $\{x_k, 1, \ldots, x_k, m\} \not\subset D$, then STOP!

**Step 2.** Set $G_k = \{\nabla f(x_k), \nabla f(x_k, 1), \ldots, \nabla f(x_k, m)\}$ and find $g_k = G_k \lambda_k$, where $\lambda_k$ solves

$$\min_{\lambda} \frac{1}{2} \lambda^T G_k^T G_k \lambda$$

s.t. $e^T \lambda = 1, \lambda \geq 0$.

**Step 3.** If $\|g_k\| \leq \nu_0$ and $\epsilon_k \leq \epsilon_0$, then terminate. Otherwise, if $\|g_k\| \leq \nu_k$, 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

$$f(x_k + t_k d_k) < f(x_k) - \beta t_k \|g_k\|^2,$$

and set $\epsilon_{k+1} = \epsilon_k$ and $\nu_{k+1} = \nu_k$.

**Step 5.** If $x_k + t_k d_k \in D$, then set $x_{k+1} = x_k + t_k d_k$. Otherwise, find

$$x_{k+1} \in B(x_k + t_k d_k, \min\{|t_k, \epsilon_k\}\|d_k\|) \cap D$$

such that $f(x_{k+1}) < f(x_k) - \beta t_k \|g_k\|^2$.

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

For a locally Lipschitz continuous function, 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

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

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

Closely related to the above set, the $\epsilon$-subdifferential set [20] is a key concept in the GS functioning. The sampled points used in GS method can be seen as an attempt to approximate the $\epsilon$-subdifferential set of $f$ at $x$ [4, Theorem 2.1].

**Definition 2** ($\epsilon$-Subdifferential set, $\epsilon$-subgradient, $\epsilon$-stationary point) The $\epsilon$-subdifferential set of $f$ at $x$ is given by

$$\partial_\epsilon f(x) := \text{co} \partial f(B(x, \epsilon)).$$
Any \( v \in \partial_\varepsilon f(x) \) is known as an \( \varepsilon \)-subgradient of \( f \) at \( x \). Moreover, if \( 0 \in \partial_\varepsilon f(x) \), then we say that \( x \) is an \( \varepsilon \)-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 locally Lipschitz continuous function \( f : \mathbb{R}^n \to \mathbb{R} \) at \( x \) in the direction \( v \in \mathbb{R}^n \) is given by

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

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

\[
f^\circ(x; v) = \max\{s^Tv : s \in \partial f(x)\}.
\]

For a more complete idea of the GS functioning, we present the nonnormalized version of the GS algorithm [31]. Since the sampled points are chosen in an uniform and independent way, one can show that the GS method, with probability one, will never stop due to Step 1 [31, Theorem 3.3]. Moreover, it is possible to show that if \( x_k \in D \), 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 matter [24]). Moreover, given the random nature of the method, nondeterministic results of convergence with probability one are expected [31].

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

\[
\min_{x \in \mathbb{R}^n} \left( f(x) := \max_{1 \leq i \leq p} \{\phi_i(x)\} \right),
\]

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, i.e., the functions \( \phi_i \) are not inputs for the method. This situation is distinct from the case in which the functions that comprise \( f \) are known. For such a case, many studies have been developed (see [13,16] and references therein).
3.1 Motivational example

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

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

then $f'(x_{0,1}) = -1, \quad f'(x_{0,2}) = 1$ and $g_0 = 0$ 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 $\epsilon_0$-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 $X^- = \{x \in \mathbb{R} : x < 0\}$ and $X^+ = \{x \in \mathbb{R} : x > 0\}$. 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

$$\min_{d,z} \quad z + \frac{1}{2} d^T d \quad \text{s.t.} \quad G^T_k d \leq z e,$$

where $z \in \mathbb{R}$ and $d \in \mathbb{R}^n$. Equivalently, considering $x_{k,0} := x_k$, the same direction $d_k$ can be obtained if we solve

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

Notice, however, that if we use the function values of each sampled point instead of $f(x_k)$ and an enriched second-order information, i.e., if we minimize in $d \in \mathbb{R}^n$ the model function

$$m^k_f(d) := \max_{1 \leq i \leq m} \left\{ f(x_{k,i}) + \nabla f(x_{k,i})^T (x_k + d - x_{k,i}) + \frac{1}{2} d^T H_k d \right\}, \quad (4)$$

we would have a better model for the function $f$ than the original in (3) (closer to a cutting-plane method). It should be mentioned that, although $f$ is a nonsmooth function, second-order type variants for nonsmooth optimization problems have presented good results [8,36]. This good behavior justifies the emergence of some GS variants that incorporate this kind of information [9–11,37] and also explains the choice to equip our new method with quasi-Newton techniques. As we shall see in the Numerical Results section, our procedure to update the matrix $H_k$ is very close to the BFGS method.
strategy. However, it is also possible to use the sampled points as a way to approximate the second-order information of the function \[9\].

The new quadratic optimization problem allows us to move when we have sampled in both “faces” of \(f\), that is, in \(X^-\) and \(X^+\). Moreover, 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 the differentiability check inside Step 5 is no longer necessary.

Unfortunately, this new quadratic programming problem comes at 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 sampling, we replace the Armijo’s line search by a trust-region procedure. Besides, aiming at a smooth problem, instead of dealing with (4), we solve at each iteration the following quadratic optimization problem,

\[
\min_{d,z} \quad z + \frac{1}{2} d^T H_k d \\
\text{s.t.} \quad \tilde{f}_k + G_k^T d \leq ze \\
\|d\|_\infty \leq \Delta_k,
\]

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 constraints, for some \(\Delta_k > 0\). Consequently, its dual optimization problem, after a changing of variables, can be viewed as

\[
\max_{\lambda, \omega} \quad \lambda^T \tilde{f}_k - \frac{1}{2} (G_k \lambda + \omega)^T H_k^{-1} (G_k \lambda + \omega) - \Delta_k \|\omega\|_1 \\
\text{s.t.} \quad \lambda^T e = 1 \\
\lambda \geq 0,
\]

where \(\lambda \in \mathbb{R}^m\) and \(\omega \in \mathbb{R}^n\) are the dual variables. Again, we stress that we do not use any information of \(f\) at \(x_k\) in (5), i.e., we do not compute \(f(x_k)\) neither \(\nabla f(x_k)\).

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 that the generated sequence of function values might not be monotone decreasing (the reason for this
choice will be better explained in the local convergence subsection). Additionally, with the same argument used for the GS method, GraFuS, with probability one, will never stop at Step 1.

Here we should clarify the value $\text{Pred}_{k,l}$ used in Algorithm 2. Recall that our model for the objective function $f$ is given by (4). Hence, it is straightforward to see that $m^k_f(0) = \max_i \left\{ f(x^l_{k,i}) + \nabla f(x^l_{k,i})^T (x_k - x^l_{k,i}) \right\}$, whereas $z_{k,l} + \frac{1}{2} d^T_{k,l} H_k d_{k,l}$ can be seen as $m^k_f(d_{k,l})$ by just recalling the equivalence between the minimization of (4) and the quadratic optimization problem (5). Therefore, $\text{Pred}_{k,l} = m^k_f(0) - m^k_f(d_{k,l})$, i.e., $\text{Pred}_{k,l}$ gives the decrease of the model, which estimates the decrease of the function $f$.

In order to guarantee the global convergence of the method, we suppose, from now on, the following assumption.

**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

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

| Glossary of notation |
|-----------------------|
| $k$: outer iteration counter |
| $l$: inner iteration counter |
| $x_k$: current iterate |
| $m$: number of sampled points |
| $\gamma_{\Delta}$: constant related to the trust region |
| $\gamma_{\epsilon}$: constant related to the sampling size |
| $\rho$: parameter of step acceptance |
| $\nu_k$: optimality certificate |
| $\nu_{\text{opt}}$: optimality certificate tolerance |
| $\vartheta$ and $\delta$: constants for updating $\nu_k$ |
| $\epsilon_{k,l}$: related to the current sampling size |
| $\Delta_{k,l}$: current trust-region size |
| $\theta$: reduction factor for $\epsilon_{k,l}$ and $\Delta_{k,l}$ |
| $\sigma_k$: power related to the sampling size |

**Remark 1** One can reason that, by the way GraFuS was designed, the trust-region size can be dramatically reduced near a stationary point of $f$ that is not a local minimizer, which could compromise the efficiency of the method to solve the problem. As remarked at the introduction of this study, the idea behind GraFuS is to replace latter iterations of the GS algorithm to obtain a faster local convergence rate, and therefore, the method was planned to run inside a neighborhood of a stationary point, previously reached by the GS method.

The updating procedure of the matrices $H_k$ is a delicate matter, since a bad sampling at one single iteration might damage some required properties for the convergence theory. For that reason, we give a detailed explanation of how one may update $H_k$ properly in Sect. 5.1.
Before we proceed with the convergence analysis, we should state a property for the functions \( \phi \) that define \( f \). It is a common assumption when we are dealing with nonsmooth functions of the kind defined in (2), cf. [12,42]. Considering that \( I(x) := \{ i : \phi_i(x) = f(x) \} \), the required hypothesis follows.

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

\[
\sum_{i \in I(x)} \alpha_i \nabla \phi_i(x) = 0 \quad \text{and} \quad \sum_{i \in I(x)} \alpha_i = 0 \iff \alpha_i = 0, \quad \text{for all} \ i \in I(x).
\]
Remark 2 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 \( |\mathcal{I}(x)| \geq 2 \) (see [26, Chapter III]). This association will be of great importance for both the global and the local convergence results.

Additionally, if \( x_\ast \) is a stationary point of \( f \), Assumption 2 also gives us that there exists only one possible convex combination of the gradients \( \nabla \phi_i(x_\ast) \), with \( i \in \mathcal{I}(x_\ast) \), that generates the null vector.

4.1 Global convergence

First, we present a technical lemma guaranteeing 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 \( \phi_j \), with \( j \in \mathcal{I}(x) \), there is a sufficiently small open set such that \( \phi_j \) strictly assumes the maximum value at this specific set.

Lemma 1 Under Assumption 2, 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 \( \epsilon > 0 \) such that for all \( \epsilon \in (0, \epsilon) \), we can find a set \( \mathcal{C}_j(x, \epsilon) \subseteq B(x, \epsilon) \) with \( \text{int}(\mathcal{C}_j(x, \epsilon)) \neq \emptyset \), for which \( x \notin \mathcal{C}_j(x, \epsilon) \) and

\[ \phi_j(y) > \max_{1 \leq i \leq n, i \neq j} \phi_i(y), \text{ for all } y \in \mathcal{C}_j(x, \epsilon). \]

Proof First, let us prove that \( |\mathcal{I}(x)| \leq n + 1 \). If \( |\mathcal{I}(x)| = 1 \), the statement trivially holds. In case \( |\mathcal{I}(x)| \geq 2 \), Assumption 2 and [26, Section III.1.3] tell us that

\[ \mathcal{A} := \{ \nabla \phi_i(x) - \nabla \phi_1(x) \}_{i \in \mathcal{I}(x) \setminus \{1\}} \]

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

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 \( |\mathcal{I}(x)| \geq 2 \). 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 \( \{v_i : i \in \mathcal{I}(x), i \neq j\} \). Thus, it is possible to find a unitary \( d_j \in \mathbb{R}^n \) such that \( v_j^T d_j = 0 \) and \( v_i^T d_j = 0, i \neq j \) with \( i \in \mathcal{I}(x) \). Consequently, it follows that \( \nabla \phi_j(x)^T d_j > \nabla \phi_s(x)^T d_j \) and \( \nabla \phi_i(x)^T d_j = \nabla \phi_s(x)^T d_j, i \neq j \) with \( i \in \mathcal{I}(x) \). So,

\[ \text{For example, setting } s_j \text{ as the orthogonal projection of } v_j \text{ over the hyperplane generated by } \{v_i : i \in \mathcal{I}(x), i \neq j\}, \text{ one can consider } d_j = (v_j - s_j)/\|v_j - s_j\|. \]
since $\phi_i \in C^2$, for all $i \in I(x)$, we have, for any fixed $w_j \in \mathbb{R}^n$ and $\epsilon > 0$, that

$$\phi_i(x + \epsilon(d_j + w_j)) = \phi_i(x) + \epsilon \nabla \phi_i(x)^T (d_j + w_j) + O(\epsilon^2), \quad i \in I(x), \quad i \neq j,$$

$$\phi_j(x + \epsilon(d_j + w_j)) = \phi_j(x) + \epsilon \nabla \phi_j(x)^T (d_j + w_j) + O(\epsilon^2).$$

Now, subtracting the first equation above from the second one and dividing the result by $\epsilon$, we obtain, for all $i \in I(x)$ with $i \neq j$, that

$$\frac{\phi_j(x + \epsilon(d_j + w_j)) - \phi_i(x + \epsilon(d_j + w_j))}{\epsilon} = \frac{\nabla \phi_j(x)^T (d_j + w_j)}{\epsilon} - \nabla \phi_i(x)^T (d_j + w_j) + O(\epsilon).$$

Consequently, supposing that $w_j \in B(0, \delta(x)) \subset \mathbb{R}^n$, where

$$\delta(x) := \min_{i \in I(x)} \left[ \min_{i \neq j} \left\{ \frac{[\nabla \phi_j(x) - \nabla \phi_i(x)]^T d_j}{2\|\nabla \phi_j(x) - \nabla \phi_i(x)\|} \right\} > 0, \quad (6)$$

we must have, for all $i \in I(x)$ with $i \neq j$, that

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

From the inequality above and noticing that $[\nabla \phi_j(x) - \nabla \phi_i(x)]^T d_j > 0$, for all $i \in I(x)$ with $i \neq j$, it is possible to find $\epsilon_j > 0$ small enough such that, for all $\epsilon \in (0, \epsilon_j)$, we have $\phi_j(x + \epsilon(d_j + w_j)) > \phi_i(x + \epsilon(d_j + w_j))$, for all $i \in I(x)$ with $i \neq j$. To complete the proof, notice that the functions $\phi_i$ are continuous, and therefore, it is possible to find $\tilde{\epsilon} > 0$ such that, for all $y \in B(x, \tilde{\epsilon})$, it follows that $\phi_a(y) > \phi_b(y)$, $a \in I(x)$ and $b \notin I(x)$. So, setting $\bar{\epsilon} := \min\{\epsilon_j, \tilde{\epsilon}\}$ and choosing $\epsilon \in (0, \bar{\epsilon})$, we have that the set

$$C_j(x, \epsilon) := \{x + \tau(d_j + w_j) : 0 < \tau < \epsilon/2, \ w_j \in B(0, \delta(x)), \ j \in I(x)\},$$

where $\delta(x)$ is the value defined in (6), satisfies the properties previously claimed. □
From the above result, we can see that, for any $\epsilon > 0$ (even when $\epsilon \geq 0$, since in this case we have $B(x, \epsilon) \subset B(x, \epsilon)$), the following set is not empty

$$S_j(x, \epsilon) := \text{int} \left\{ y \in B(x, \epsilon) : \phi_j(y) > \max_{1 \leq i \leq p \atop i \neq j} \phi_i(y) \right\}, \ j \in \mathcal{I}(x). \quad (7)$$

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 5 to Step 1. Specifically, the first result tells us that under a good set of sampled points, it is possible to obtain $A_{\text{red}} > \rho_{\text{Pred}}$ (the proof of the result is based on ideas from [52]).

**Lemma 2** Suppose Assumptions 1 and 2 hold, and that $\bar{x} \in \mathbb{R}^n$ is a nonstationary point for the function $f : \mathbb{R}^n \to \mathbb{R}$. Moreover, let $\rho \in (0, 1)$ be a fixed real number, $M$ be any positive real value and $S_j(\bar{x}, \epsilon)$ be the set defined in (7) for any $\epsilon > 0$. Therefore, there exist $\Delta$ and $\delta$ strictly greater than zero such that, if the following hypotheses hold

(i) $x_k \in B(\bar{x}, \delta)$;
(ii) $0 < \Delta_{k,l} < \Delta$;
(iii) there exists $\bar{\epsilon} \equiv \bar{\epsilon}(k, l) > 0$ such that

(a) for all $j \in \mathcal{I}(\bar{x})$, we have $S_j(\bar{x}, \epsilon) \subset B(x_k, M \cdot \Delta_{k,l})$;
(b) for all $j \in \mathcal{I}(\bar{x})$, there exists $i \in \{1, \ldots, m\}$ such that $x_{k,i}^j \in S_j(\bar{x}, \epsilon)$;
(c) for all $i \in \{1, \ldots, m\}$, there exists $j \in \mathcal{I}(\bar{x})$ such that $x_{k,i}^j \in S_j(\bar{x}, \epsilon)$,

for any fixed (outer, inner) iteration $(k, l)$ in Algorithm 2, then $A_{\text{red}}(k, l) > \rho_{\text{Pred}}(k, l)$.

**Proof** First, we choose $h > 0$ as a sufficiently small number such that for all $\|x - \bar{x}\|_\infty \leq 2h$, we have

$$\phi_j(x) > \max_{1 \leq i \leq p \atop i \neq j} \phi_i(x), \text{ for all } j \in \mathcal{I}(\bar{x}).$$

Since $\bar{x}$ is not a stationary point for $f$, we must have $0 \notin \partial f(\bar{x})$. Recalling that $\partial f(\bar{x})$ 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

$$s^T v \leq -\tau, \text{ for all } s \in \partial f(\bar{x}).$$

Since the generalized directional derivative of $f$ at $\bar{x}$ in the direction $v$ is given by

$$f^\circ(\bar{x}; v) = \limsup_{x \to \bar{x}} \lim_{t \downarrow 0} \frac{f(x + tv) - f(x)}{t} = \max \{ s^T v : s \in \partial f(\bar{x}) \},$$

\( \copyright \) Springer
we have that \( f^o(X;v) \leq -\tau \). Thus, there exist \( \Delta \in (0, h) \) and \( \delta \in (0, h) \) such that for all \( x \in B(\overline{X}, \overline{\delta}) \) and \( \Delta \in (0, \overline{\Delta}) \), we have

\[
f(x + \Delta v) - f(x) < -\frac{\tau}{2} \Delta.
\]

(8)

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 \( \delta \) and \( \overline{\Delta} \) found above. Then, by the conditions in \( iii) \), we have, due to \( x^l_{k,i} \in B(x_k, M \cdot \Delta_{k,l}) \) and \( \|d_{k,l}\|_\infty \leq \Delta_{k,l} \), that

\[
f(x_k) = \max_{j \in J(x)} \{ \phi_j(x_k) \} = \max_{1 \leq i \leq m} \{ f(x^l_{k,i}) + \nabla f(x^l_{k,i})^T (x_k - x^l_{k,i}) \} + o(\Delta_{k,l}) \tag{9}
\]

and

\[
f(x_k + d_{k,l}) = \max_{j \in J(x)} \{ \phi_j(x_k + d_{k,l}) \} = \max_{1 \leq i \leq m} \{ f(x^l_{k,i}) + \nabla f(x^l_{k,i})^T (x_k + d_{k,l} - x^l_{k,i}) \} + o(\Delta_{k,l}).
\]

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 \( \hat{z} := \max_{1 \leq i \leq m} \{ f(x^l_{k,i}) + \nabla f(x^l_{k,i})^T (x_k + \Delta_{k,l} v - x^l_{k,i}) \} \).

Notice that, by the same reasoning used before, we have

\[
\hat{z} = f(x_k + \Delta_{k,l} v) + o(\Delta_{k,l}). \tag{10}
\]

Moreover, since \( (d_{k,l}, z_{k,l}) \) is the solution of the quadratic programming problem in Step 2, we have \( z_{k,l} \leq \hat{z} + o(\Delta_{k,l}) \), and hence,

\[
\text{Pred}_{k,l} \geq \max_{1 \leq i \leq m} \{ f(x^l_{k,i}) + \nabla f(x^l_{k,i})^T (x_k - x^l_{k,i}) \} - \left( \hat{z} + \frac{\Delta_{k,l}^2}{2} v^T H_k v \right) + o(\Delta_{k,l}).
\]

Consequently, recalling (9) and (10), it yields that

\[
\text{Pred}_{k,l} \geq f(x_k) - f(x_k + \Delta_{k,l} v) + o(\Delta_{k,l}) > \frac{\tau}{2} \Delta_{k,l} + o(\Delta_{k,l}),
\]

where the last inequality comes from (8). Therefore, if \( \overline{\Delta} \) is small enough, we obtain the desired result. \( \square \)
With the above result, we present the following lemma, which states that if GraFuS is in an 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 Assumptions 1 and 2 hold. Moreover, for an iteration \( k \), assume that \( x_k \) is not a stationary point for \( f \). Then, with probability one, there exists \( \tilde{l} \in \mathbb{N} \) such that the indices of the inner iterations satisfy \( l \leq \tilde{l} \).

**Proof** Let us assume, for contradiction, that such \( \tilde{l} \) does not exist, i.e., \( l \to \infty \) at the iteration \( k \). Consequently, we must have, for all \( l \in \mathbb{N} \), that \( \| H^{-1}_k G_{k,l} \lambda_{k,l} \| \geq \nu_k \) and \( \text{Ared}_{k,l} \leq \rho \text{Pred}_{k,l} \). Additionally, by the way we have designed our algorithm, we see that \( \epsilon_{k,l} = (\gamma_{\epsilon} / \gamma_{\Delta}) \Delta_{k,l} \) for all \( l \in \mathbb{N} \), and, by the contradiction hypothesis, the following holds: \( \Delta_{k,l} \to 0 \) as \( l \to \infty \).

Therefore, setting \( x := x_k \) in Lemma 2, it is straightforward to see that at some \( \tilde{n} \in \mathbb{N} \), if \( l \geq \tilde{n} \), then hypotheses \( i \) and \( ii \) of Lemma 2 are valid. Moreover, considering \( \bar{\epsilon} := (\epsilon_{k,l})^{\sigma_k} \) and \( M := \max \{ \gamma_{\epsilon}^{-1} \gamma_{\Delta} \} \) for a fixed inner iteration \( l \), we will satisfy hypothesis \( iii \) item \( a \) of Lemma 2. Therefore, since \( \text{Ared}_{k,l} \to \rho \text{Pred}_{k,l} \) does not hold, it must follow 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 \( C_j(x_k, (\epsilon_{k,l})^{\sigma_k}) \) in the proof of Lemma 1, we have that (for \( (\epsilon_{k,l})^{\sigma_k} \) sufficiently small) \( B^{k,l}_j \subset C_j(x_k, (\epsilon_{k,l})^{\sigma_k}) \), where

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

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

\[
\text{Vol} \left( B^{k,l}_j \right) = \frac{\pi^{n/2}}{\Gamma(n/2 + 1)} \left( \min_{i \in \mathcal{I}(x_k)} \left\{ \frac{[\nabla \phi_j(x_k) - \nabla \phi_i(x_k)]^T d_j}{2\| \nabla \phi_j(x_k) - \nabla \phi_i(x_k) \|} \right\} \right)^n \left( \frac{(\epsilon_{k,l})^{\sigma_k}}{8} \right)^n,
\]

where \( \Gamma \) is the Gamma function [27]. On the other hand, it follows that

\[
\text{Vol}(B(x_k, (\epsilon_{k,l})^{\sigma_k})) = \frac{\pi^{n/2}}{\Gamma(n/2 + 1)} ((\epsilon_{k,l})^{\sigma_k})^n.
\]

Therefore, since the sampled points are chosen in \( B(x_k, (\epsilon_{k,l})^{\sigma_k}) \) and

\[
B^{k,l}_j \subset C_j(x_k, (\epsilon_{k,l})^{\sigma_k}) \subset S_j(x_k, (\epsilon_{k,l})^{\sigma_k}),
\]

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

$$
P(x_{k,i}^l \in S_j(x_k, (\epsilon_{k,l})^{\sigma_k}) \mid x_{k,i}^l \in B(x_k, (\epsilon_{k,l})^{\sigma_k})) = \frac{\text{Vol}(S_j(x_k, (\epsilon_{k,l})^{\sigma_k}))}{\text{Vol}(B(x_k, (\epsilon_{k,l})^{\sigma_k}))}
$$

must be greater than the following strictly positive number

$$
\frac{1}{8^n} \left( \min_{i \in \mathcal{I}(x_k)} \frac{\| \nabla \phi_j(x_k) - \nabla \phi_i(x_k) \|_1}{2 \| \nabla \phi_j(x_k) - \nabla \phi_i(x_k) \|_2} \right)^n.
$$

With this inequality, we conclude that the probability of the items $b)$ and $c)$ of hypothesis $iii)$ to occur simultaneously is strictly positive and does not depend on $l$. Therefore, the probability of $l \to \infty$ is zero, which concludes the proof. \qed

We are close to reach the convergence theorem of GraFuS. For that goal, we need to prove two additional technical lemmas. Furthermore, to have a clearer proof, from now on we will denote by $\tilde{l}_k$ the largest value of the index $l$ in the iteration $k$, established by Lemma 3.

**Lemma 4** Let us consider the GraFuS algorithm under Assumptions 1 and 2. If there exists an infinite index set $\tilde{K} \subset \mathbb{N}$ such that $\text{Pred}_{k,\tilde{l}_k}/\Delta_{k,\tilde{l}_k} \to 0$, then $\| G_{k,\tilde{l}_k} \gamma_{k,\tilde{l}_k} \| \to 0$.

**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

$$
z_{k,\tilde{l}_k} + \frac{1}{2} d_{k,\tilde{l}_k}^T H_k d_{k,\tilde{l}_k} = \lambda_{k,\tilde{l}_k}^T \tilde{f}_{k,\tilde{l}_k} - \frac{1}{2} \left( G_{k,\tilde{l}_k} \gamma_{k,\tilde{l}_k} + \omega_{k,\tilde{l}_k} \right)^T H_k^{-1} \left( G_{k,\tilde{l}_k} \gamma_{k,\tilde{l}_k} + \omega_{k,\tilde{l}_k} \right) - \Delta_{k,\tilde{l}_k} \| \omega_{k,\tilde{l}_k} \|_1.
$$

Thus, defining

$$
\alpha_k := \frac{1}{2} \left( G_{k,\tilde{l}_k} \gamma_{k,\tilde{l}_k} + \omega_{k,\tilde{l}_k} \right)^T H_k^{-1} \left( G_{k,\tilde{l}_k} \gamma_{k,\tilde{l}_k} + \omega_{k,\tilde{l}_k} \right) - \Delta_{k,\tilde{l}_k} \| \omega_{k,\tilde{l}_k} \|_1, \quad (11)
$$
it yields

\[
\lambda_{k,\bar{I}_k}^T \tilde{f}_{k,\bar{I}_k} - \alpha_k = z_{k,\bar{I}_k} + \frac{1}{2} d_{k,\bar{I}_k}^T H_{k,\bar{I}_k} d_{k,\bar{I}_k} \Rightarrow \alpha_k = \lambda_{k,\bar{I}_k}^T \tilde{f}_{k,\bar{I}_k} - \left( z_{k,\bar{I}_k} + \frac{1}{2} d_{k,\bar{I}_k}^T H_{k,\bar{I}_k} d_{k,\bar{I}_k} \right)
\]

\[
\Rightarrow \alpha_k \leq \text{Pred}_{k,\bar{I}_k}
\]

(since \(\lambda_{k,\bar{I}_k} \geq 0\) and \(e^T \lambda_{k,\bar{I}_k} = 1\))

\[
\Rightarrow \frac{\alpha_k}{\Delta_{k,\bar{I}_k}} \leq \frac{\text{Pred}_{k,\bar{I}_k}}{\Delta_{k,\bar{I}_k}}
\]

\[
\Rightarrow \Delta_{k,\bar{I}_k} \rightarrow 0.
\]

Consequently, by Assumption 1 and (11), we obtain \(\|G_{k,\bar{I}_k} \lambda_{k,\bar{I}_k}\| \rightarrow 0\) for \(k \in \bar{K}\). \(\square\)

Finally, we present the last result before our main statement of the global convergence analysis.

**Lemma 5** Suppose that Assumptions 1 and 2 hold and GraFuS has generated an infinite sequence \(\{x_k\} \subset \mathbb{R}^n\). Moreover, assume that there exists a cluster point \(\bar{x}\) of this sequence that is a stationary point for \(f\). Then, with probability one, the sequence \(\{v_k\}\) must converge to zero.

**Proof** By hypothesis, we have that \(0 \in \overline{\partial} f(\bar{x})\). Moreover, all the functions that comprise \(f\) are of class \(C^2\). So, it is possible to find, for any given \(\delta_1, \delta_2 > 0\), nonempty and open sets \(\mathcal{X}_1, \ldots, \mathcal{X}_m \subset \mathcal{D}\) and a fixed vector \(\bar{\lambda} \in \mathbb{R}^m\) satisfying \(\bar{\lambda} \geq 0\) and \(e^T \bar{\lambda} = 1\) such that

\[
\mathcal{X}_j \subset B(\bar{x}, \delta_1), \text{ for all } j \in \{1, \ldots, m\},
\]

and

\[
\left\| \sum_{j=1}^{m} \bar{\lambda}_j \nabla f(x_j) \right\| \leq \delta_2, \text{ for all } (x_1, \ldots, x_m) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_m.
\]

By contradiction, let us assume that \(\{v_k\}\) does not go to zero, i.e., there exists \(\overline{v} > 0\) such that \(v_k = \overline{v}\) for all \(k \in \mathbb{N}\) sufficiently large. This condition yields that \(\epsilon_{k,0} = \gamma_{\epsilon} \overline{v}\) and \(\Delta_{k,0} = \gamma_{\Delta} \overline{v}\) for all \(k \in \mathbb{N}\) large enough. Moreover, noticing that Lemma 4 also holds if we consider the inner iteration 0 instead of \(\bar{I}_k\), we have that

\[
\frac{\text{Pred}_{k,0}}{\Delta_{k,0}} \geq \mu,
\]

for some \(\mu > 0\). Otherwise, \(\|G_{k,0} \lambda_{k,0}\|\) would go to zero, implying that \(v_k\) would also go to zero.

\(\square\) Springer
Defining $\mathcal{K}$ as an infinite index set such that $\{x_k\}_{k \in \mathcal{K}}$ converges to $\bar{x}$, it is possible to find $\delta_1$ small enough such that (12) holds and

$$\mathcal{X}_j \subset B(x_k, (\epsilon, 0))_{\sigma_k}, \text{ for all } j \in \{1, \ldots, m\} \text{ and } k \in \mathcal{K} \text{ large enough.} \tag{15}$$

So, let us suppose that for some $k \in \mathcal{K}$ sufficiently large, we have $x_k^0, j \in \mathcal{X}_j$, for all $j \in \{1, \ldots, m\}$. Then, considering $\lambda_{k,0}$ and $\omega_{k,0}$ the solutions obtained in Step 2, we must have

$$\lambda^T \tilde{f}_{k,0} - \frac{1}{2} \lambda^T G_{k,0} H_k^{-1} G_{k,0} \bar{x} \leq \lambda^T \tilde{f}_{k,0} - \frac{1}{2} \left(G_{k,0} \lambda_{k,0} + \omega_{k,0}\right)^T H_k^{-1} \left(G_{k,0} \lambda_{k,0} + \omega_{k,0}\right) - \Delta_{k,0} \|\omega_{k,0}\|_1. $$

Adding $\max_{1 \leq j \leq m} \left\{ \tilde{f}_{k,0} \right\}_j$ to both sides of the inequality that comes from multiplying the previous one by $(-1)$ and considering the strong duality of the quadratic problem that is solved in Step 2, we have

$$\text{Pred}_{k,0} \leq \max_{1 \leq j \leq m} \left\{ \tilde{f}_{k,0} \right\}_j - \lambda^T \tilde{f}_{k,0} + \frac{1}{2} \lambda^T G_{k,0} H_k^{-1} G_{k,0} \lambda.$$

Since $f(\bar{x}) = \phi_i(\bar{x})$, for any $i \in I(\bar{x})$, it is possible to select a sufficiently small $\delta_1$, such that (12), (15) and

$$\max_{1 \leq j \leq m} \left\{ \tilde{f}_{k,0} \right\}_j \leq \mu \frac{\gamma \Delta V}{4},$$

are valid for any $k \in \mathcal{K}$ large enough. Moreover, by Assumption 1, it is possible to choose $\delta_2$ sufficiently small such that (13) holds and

$$\frac{1}{2} \lambda^T G_{k,0} H_k^{-1} G_{k,0} \lambda \leq \mu \frac{\gamma \Delta V}{4},$$

for any $k \in \mathcal{K}$ large enough.

As a result, there are $\delta_1, \delta_2 > 0$ sufficiently small and $k \in \mathcal{K}$ sufficiently large, such that, if $x_k^0, j \in \mathcal{X}_j$, for all $j \in \{1, \ldots, m\}$, we have

$$\frac{\text{Pred}_{k,0}}{\Delta_{k,0}} \leq \mu \frac{\gamma \Delta V}{2 \Delta_{k,0}} = \mu \frac{\gamma \Delta V}{2 \Delta_{k,0}} = \mu \frac{\gamma \Delta V}{2}.$$

Since we have supposed that $\nu_k$ does not go to zero, it implies that GraFuS never samples in the nonempty and open set $\mathcal{X}_1 \times \cdots \times \mathcal{X}_m$ during the iterations $k \in \mathcal{K}$, since, otherwise, we would have a contradiction with (14). This is an event that has probability zero to occur. Therefore, with probability one, the sequence $\{\nu_k\}$ must converge to zero. \qed
Now, we present the main result of this subsection. The following theorem assumes that \( f \) has bounded level sets, but it is sufficient to ask the sequence \( \{x_k\} \) produced by GraFuS to be bounded. We have chosen the former over the latter, because this allows us to avoid an algorithmic hypothesis. Although the GS method does not require such an assumption [31], this seems necessary in our case due to the nonmonotone behavior of the proposed method.

**Theorem 1** Under Assumptions 1 and 2, suppose that \( f \) has bounded level sets and GraFuS produces an infinite sequence \( \{x_k\} \) with \( v_{\text{opt}} = 0 \). Then, with probability one, the sequence \( \{v_k\} \) converges to zero.

**Proof** We split the proof in two complementary cases:

(i) There are an infinite set of indices \( K_1 \subset \mathbb{N} \) and a real number \( \epsilon > 0 \) such that \( \epsilon_k \geq \epsilon \) for all \( k \in K_1 \).

(ii) The sampling radius along the iterations satisfy \( \epsilon_k \to k \in \mathbb{N} \).

Initially, let us suppose that case 1) holds. So, noticing that \( \epsilon_k \leq \gamma \epsilon \nu_k \) for all \( k \in \mathbb{N} \), and that \( \{v_k\} \) is a monotonically decreasing sequence, we see clearly that there must exist \( \nu \) such that \( \nu_k \geq \nu \), for all \( k \in K_1 \). Therefore, by the way GraFuS was designed, there must exist \( \mu > 0 \) such that \( \Delta_k \mu \leq \nu_k \) for all \( k \in \mathbb{N} \). Indeed, if this statement were false, there would exist an infinite set of indices \( \tilde{K} \) such that \( \nu_k \to k \in \mathbb{N} \). However, by Lemma 4, it would yield \( \|G_k_{\lambda}\| \to k \in \mathbb{N} \). Therefore, we have

\[
A_{\text{red}} \geq \rho \frac{\gamma \Delta}{\gamma \epsilon} \epsilon, \quad \text{for all } k \in \mathbb{K}.
\]

we obtain

\[
f(x_k) - f(x_{k+1}) > \rho \frac{\gamma \Delta}{\gamma \epsilon} \epsilon, \quad \text{for all } k \in \mathbb{K}.
\]

Now, since \( f \) has bounded level sets, there must exist an infinite set of indices \( \mathbb{K}_2 \subset \mathbb{K}_1 \) such that

\[
x_k \to \hat{x}, \quad \text{for some } \hat{x} \in \mathbb{R}^n.
\]

So, considering \( s_{\mathbb{K}_2}(k) \) as the index in \( \mathbb{K}_2 \) that comes right after \( k \in \mathbb{K}_2 \) and recalling that, for the case at hand, it is possible to find a sufficiently large \( \hat{k} \in \mathbb{K}_2 \), where the

\[ \widehat{\text{Springer}} \]
A fast gradient and function sampling method for…

sequence of function values will be a decreasing sequence for all \( k \in \mathbb{N} \) and \( k \geq \hat{k} \), it yields that

\[
\sum_{k \in \mathcal{K}_2, k \geq \hat{k}} (f(x_k) - f(x_{k+1})) \leq \sum_{k \in \mathcal{K}_2, k \geq \hat{k}} \left( f(x_k) - f(x_{\mathcal{K}_2(k)}) \right) = f(x_{\hat{k}}) - f(\tilde{x}) < \infty.
\]

However, this is a relation that goes against (16). Therefore, the case \( \text{i) is an impossible event and we must consider case } \text{ii).}

So, suppose that case \( \text{ii) holds and, by contradiction, that the sequence } \{\nu_k\} \text{ does not converge to zero. Again, we must have that the inequality } \|H_{k,l}^{-1} G_{k,l} \| < \nu_k \text{ will never hold for } k \text{ sufficiently large, and consequently, the sequence generated by the values } f(x_k) \text{ will decrease monotonically. Thus, there must exist at least one cluster point } \bar{x} \text{ of } \{x_k\}. \text{ Consequently, there is } \hat{\mathcal{K}} \subset \mathbb{N} \text{ such that } x_k \to \hat{x}. \text{ Now, because of Lemma 5, } \bar{x} \text{ is not a stationary point for } f. \text{ Then, we choose } \delta, \overline{\Delta} \text{ as presented in Lemma 2 for the point } \bar{x}. \text{ Since } v_k \text{ remains bounded away from zero by our assumption and } \epsilon_k, l_k \to 0, \text{ we have, by the way we have designed GraFuS, that } \epsilon_k, l_k \text{ just keeps going smaller because } l_k \to \infty. \text{ As a consequence, there exist } k', l' \in \mathbb{N} \text{ such that for all } k \geq k' \text{ we have }

\[
\Delta_{k,l'} = \overline{\Delta} := \left( \theta^{l'} \right) \gamma_{\Delta} v_k < \overline{\Delta} \text{ and } \epsilon_k, l' = \tilde{\epsilon} := \left( \theta^{l'} \right) \gamma_{\epsilon} v_k = \frac{\gamma_{\epsilon}}{\gamma_{\Delta}} \overline{\Delta}.
\]

Moreover, since \( \bar{x} \) is a cluster point for the sequence of iterates, we can find \( \hat{k} \geq k' \) such that for all \( k \geq \hat{k} \) and \( k \in \hat{\mathcal{K}} \), we have

\[
x_k \in B(\bar{x}, \min\{\tilde{\epsilon}^2, \delta\}/4) \subset B(\bar{x}, \min\{\tilde{\epsilon} \sigma_k, \delta\}/4)
\]

So, for all \( j \in \mathcal{I}(\bar{x}) \), we have

\[
x_k \in B(\bar{x}, \min\{\tilde{\epsilon} \sigma_k, \delta\}/4) \text{ and } \mathcal{S}_j(\bar{x}, \min\{\tilde{\epsilon}, \delta\}/4) \subset B\left(x_k, \frac{\gamma_{\epsilon}}{\gamma_{\Delta}} \overline{\Delta} \right).
\]

Therefore, the hypotheses \( \text{i), ii) and iii) item a) of Lemma 2 are all satisfied. Thus, since } \mathcal{I}_k \to \infty, \text{ we must have that items b) and/or c) of hypothesis } \text{iii) are not satisfied for every } k \geq \hat{k} \text{ and } l = l'. \text{ However, this is an event with probability zero of happening, since the sets } \mathcal{S}_j(\bar{x}, \min\{\tilde{\epsilon}, \delta\}/4) \text{ are open and not empty. Consequently, with probability one, the sequence } \{\nu_k\} \text{ must converge to zero.} \]

In the light of the above theorem, the next corollary ensures that GraFuS will find, in a finite number of iterations, an \( \epsilon \)-stationary point under any given tolerance. Furthermore, it justifies calling \( v_k \) an optimality certificate.

**Corollary 1** Under Assumptions 1 and 2, suppose that \( f \) has bounded level sets and the parameter value \( v_{opt} \) in GraFuS is strictly positive. Then, with probability one,
GraFuS terminates in a finite number of iterations. Moreover, there exists $v \in \mathbb{R}^n$ such that

$$v \in \partial \tilde{f}(x_{\hat{k}}) \text{ with } \|v\| \leq \tilde{v} := \overline{\sigma} \cdot v_{\hat{k}},$$

where $\hat{k}$ is the final iteration of GraFuS, $\tilde{\epsilon} := \gamma \epsilon v_{\hat{k}}$ and $\overline{\sigma}$ is the constant presented in Assumption 1. In other words, $x_{\hat{k}}$ is an $\tilde{\epsilon}$-stationary point under the tolerance $\tilde{v}$.

**Proof** The proof follows immediately from Theorem 1, Assumption 1 and by the way GraFuS was designed.

The next result guarantees that, if the sequence $\{x_k\}$ produced by GraFuS is bounded, then we also obtain an asymptotic result.

**Corollary 2** Under Assumptions 1 and 2, suppose that GraFuS produces an infinite and bounded sequence $\{x_k\}$ with $v_{\text{opt}} = 0$. Then, with probability one, there exist an infinite index set $\mathcal{K} \subset \mathbb{N}$ and $x_* \in \mathbb{R}^n$ such that $x_k \to_{k \in \mathcal{K}} x_*$, $x_*$ is a stationary point for $f$, and $v_{k+1} < v_k$, for all $k \in \mathcal{K}$.

**Proof** The result follows immediately from Theorem 1. Notice that replacing the boundedness of the level sets by the boundedness of $\{x_k\}$ does not invalidate the proof of Theorem 1. So, defining the infinite index set $\tilde{\mathcal{K}} := \{k \in \mathbb{N} : v_{k+1} < v_k\}$, since $\{x_k\}$ is bounded, there must exist a point $x_* \in \mathbb{R}^n$ and an infinite index set $\mathcal{K} \subset \tilde{\mathcal{K}}$ such that $x_k \to_{k \in \mathcal{K}} x_*$. Therefore, since $v_k \to 0$, and there exists $v_k \in \mathbb{R}^n$ such that $v_k \in \partial (\tilde{\epsilon} f(x_k))$ with $\|v_k\| \leq \overline{\sigma} \cdot v_k$, for any $k \in \mathcal{K}$, we have the desired result (see item iii) of [31, Lemma 3.2]), i.e., $0 \in \partial f(x_*)$ with probability one.

It should be stressed that, with probability one, any cluster point associated with the subsequence of iterates related to the index set $\tilde{\mathcal{K}} = \{k \in \mathbb{N} : v_{k+1} < v_k\}$ is a stationary point of $f$.

In the next subsection, we show that, under a good sampling, the method super-linearly either moves to a local minimizer of $f$ or reduces the optimality certificate. For such a goal, our analysis will involve the concept of $\mathcal{VU}$-spaces (see Definition 4 below).

### 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 $\phi_i$. Under 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. Additionally, the
The following definition presents key concepts for our analysis (a more general definition can be found in [35]).

**Definition 4 (U-spaces)** Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is the continuous objective function of problem (2) and $x$ is any point in $\mathbb{R}^n$ with $I(x) \geq 2$. Then, we define

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

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

We have seen in Corollary 2 that, under reasonable hypotheses, GraFuS will generate a cluster point $x_*$ which is stationary for $f$. In order to proceed with our analysis, we assume, along this entire subsection, that $x_* \in \mathbb{R}^n$ is a local minimizer of the optimization problem presented in (2), and that $x_*$ is also a strong minimizer for $f$ [43, Section 5.1].

To accomplish the aim of this subsection, under Assumption 2, we start supposing, without any loss of generality, that $I(x_*) = \{1, \ldots, r + 1\}$, for some $r \leq n$.

**Assumption 3** The local minimizer $x_*$ of problem (2) is a strong minimizer, i.e., $0 \in \partial f(x_*)$ and there exists $\mu > 0$ such that

$$d^T \left(\sum_{i=1}^{r+1} (\lambda_*)_i \nabla^2 \phi_i(x_*)\right) d \geq \mu \|d\|^2, \quad \text{for all } d \in U(x_*),$$

where $\lambda_* \in \mathbb{R}^{r+1}$ is the unique vector such that

$$\lambda_* \geq 0, \quad \sum_{i=1}^{r+1} (\lambda_*)_i = 1 \quad \text{and} \quad \sum_{i=1}^{r+1} (\lambda_*)_i \nabla \phi_i(x_*) = 0.$$

Below, we present our first technical result that characterizes a good set of sampled points, and it will prove helpful for the subsequent statements. We say that a good sampling has been generated, when the sampled points satisfy item i) and the relation a) of item ii). Part b) of item ii) is a consequence of the result, and it indicates the importance of a good sampling for the solution of the quadratic programming problem that is solved in Step 2.

**Lemma 6** Suppose that Assumptions 1, 2 and 3 hold and $\{x_k\}$ is an infinite bounded sequence generated by GraFuS with $v_{opt} = 0$. Then, with probability one, the index set $K \subset \mathbb{N}$ presented in Corollary 2 exists and the following holds for any fixed, and large enough $k \in K$:

1. For each $j \in \{1, \ldots, m\}$, there exists $i \in I(x_*)$, such that

$$\nabla f((x_{k,j}^I_1) = \nabla \phi_i((x_{k,j}^I_1).$$
(ii) for each $i \in \mathcal{I}(x_*)$, there exists $j \in \{1, \ldots, m\}$, such that

(a) $\nabla f \left( x_{k,j}^l \right) = \nabla \phi_i \left( x_{k,j}^l \right)$;

(b) $\left( \lambda_{k,l}^l \right)_j > 0$, i.e., the constraint

$$f \left( x_{k,j}^l \right) + \nabla f \left( x_{k,j}^l \right)^T \left( x_k + d - x_{k,j}^l \right) \leq z$$

is active at the optimal solution of the quadratic programming that is solved in Step 2.

**Proof** From Corollary 2, it is straightforward to see that $\mathcal{K}$ does exist with probability one. Now, let us prove items $i)$ and $ii)$.

Recalling that $\epsilon_{k,0} \to 0$ (since $\nu_k \to 0$), $x_k \to x_*$ and $\phi_i$ are all continuous functions, then, for any large outer iteration $k \in \mathcal{K}$, there exists a neighborhood $\mathcal{W} \subset \mathbb{R}^n$ of $x_*$ such that $x_{l,j}^k \in \mathcal{W}$, for all $j \in \{1, \ldots, m\}$ and $l \in \mathbb{N}$, and only the functions $\phi_i$, with $i \in \mathcal{I}(x_*)$, assume the maximum in $\mathcal{W}$, which gives us $i)$.

Now, by the way we have designed Step 3 and because $\nu_{k+1} < \nu_k$ for all $k \in \mathcal{K}$, there exists $\hat{\lambda} \in \mathbb{R}^m$ such that

$$\| H_k^{-1} G_{k,l} \hat{\lambda} \| < \nu_k, \text{ for any } k \in \mathcal{K}. \quad (18)$$

Additionally, because we assume that $0 \in \text{ri } \partial f(x_*)$ and Assumption 2 holds, it follows, by [26, Remark III.2.1.4], that

$$\sum_{i \in \mathcal{I}(x_*)} \lambda_i \nabla \phi_i(x_*) = 0 \Rightarrow \lambda_i > 0, \text{ for all } i \in \mathcal{I}(x_*). \quad (19)$$

Therefore, since the functions that comprise $f$ are assumed to be of class $C^2$, it is not possible to have (18) and $\nu_k \to 0$ without having $ii)$, item $a)$.

Now, let us prove $\| H_k^{-1} G_{k,l} \lambda_{k,l} \kappa \| \to 0$. Suppose, for contradiction, that there exist $M > 0$ and an infinite index set $\mathcal{K}_1 \subset \mathcal{K}$ such that $\| H_k^{-1} G_{k,l} \lambda_{k,l} \kappa \| \geq M$ for all $k \in \mathcal{K}_1$. Moreover, recall that, since the trust-region constraints are not active for the (outer, inner) iteration pair $(k, l_k)$ whenever $k \in \mathcal{K}$, the vector $\lambda_{k,l_k}$ is the optimal solution of

$$\max_{\lambda \in \mathbb{R}^m} \lambda^T \tilde{f}_{k,l_k} - \frac{1}{2} \lambda^T G_{k,l_k}^T H_k^{-1} G_{k,l_k} \lambda$$

s.t. $\lambda^T e = 1, \lambda \geq 0$. \quad (20)

However, any $\hat{\lambda} \in \mathbb{R}^m$ with $\| H_k^{-1} G_{k,l_k} \hat{\lambda} \| < \nu_k$ (which does exist because of (18)), will give a better objective value for problem (20) than $\lambda_{k,l_k}$, whenever $k \in \mathcal{K}_1$ is large enough (i.e., $\nu_k$ is small enough). This fact reveals an absurd. Therefore, we obtain $\| H_k^{-1} G_{k,l_k} \lambda_{k,l_k} \| \to 0$. This last information together with (19) and Assumption 1 give us relation $b)$ of item $ii)$.

$\square$
Along this subsection, every time we refer to the set written as $\mathcal{K}$, we are referring to the set $\mathcal{K}$ presented in Corollary 2, and we also suppose that the smallest index in $\mathcal{K}$ is large enough in order to items $i)$ and $ii)$ of Lemma 6 to hold for all $k \in \mathcal{K}$. Moreover, recalling the result obtained above and rearranging properly the sampled points, we can suppose, without any loss of generality, that

$$\nabla f \left( x_{k,i}^k \right) = \nabla \phi_i \left( x_{k,i}^k \right) \quad \text{and} \quad \left( \lambda_{k,i}^k \right) > 0, \quad \text{for all } k \in \mathcal{K} \text{ and } i \in \mathcal{I}(x_*) \text{.}$$

Additionally, for the sake of simplicity, we assume from now on that, for $k \in \mathcal{K}$,

$$\lambda_{k,i}^k = 0, \quad \text{if } i \notin \mathcal{I}(x_*) \text{.}$$

So, for any $k \in \mathcal{K}$, one can rewrite (5) as the following optimization problem

$$\min_{d,z} z + \frac{1}{2} d^T H_k d \quad \text{s.t.} \quad \phi_i \left( x_{k,i}^k \right) + \nabla \phi_i \left( x_{k,i}^k \right)^T \left( x_k + d - x_{k,i}^k \right) = z, \quad 1 \leq i \leq r + 1 \text{.}$$

(21)

Alternatively, it can also be viewed as

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

(22)

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

$$\left( \tilde{\Phi}_k \right)_i := \phi_i \left( x_{k,i}^k \right) + \nabla \phi_i \left( x_{k,i}^k \right)^T \left( x_k - x_{k,i}^k \right)$$

$$- \left[ \phi_{r+1} \left( x_{k,r+1}^k \right) + \nabla \phi_{r+1} \left( x_{k,r+1}^k \right)^T \left( x_k - x_{k,r+1}^k \right) \right], \quad i \in \{1, \ldots, r\} \text{,}$$

and

$$\tilde{J}_k := \begin{pmatrix}
\nabla \phi_1 \left( x_{k,1}^k \right)^T - \nabla \phi_{r+1} \left( x_{k,r+1}^k \right)^T \\
\vdots \\
\nabla \phi_r \left( x_{k,r}^k \right)^T - \nabla \phi_{r+1} \left( x_{k,r+1}^k \right)^T
\end{pmatrix} \text{.}$$

So, the minimization problem (5) can be viewed as a quadratic approximation of

$$\min_{x \in \mathbb{R}^n} \phi_{r+1}(x) \quad \text{s.t.} \quad \Phi(x) = 0 \text{,}$$

(23)
where

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

With this initial analysis, we are ready to understand why we have chosen to design a method that produces a sequence of function values that is not monotonically decreasing. When one tries to move superlinearly to a solution of a smooth constrained optimization problem, the Maratos effect [40,45] must be taken into consideration. Sometimes, a good movement towards $x^*$ might be not accepted because the candidate for the next iterate does not improve the function value. Normally, a correction step is made to prevent this undesirable property to happen and the superlinear convergence can be assured.

We have seen above that the quadratic problem that is solved in Step 2 can be seen as a smooth constrained optimization problem and one might expect that we can do the same correction step to ensure a superlinear movement towards the solution. However, since we suppose that we do not know the functions $\phi_i$, such a correction becomes very hard to perform. One could try to numerically approximate $\tilde{J}_k$ during the execution of the algorithm to create a correction step, but this estimation can be very tricky. For these reasons, we have chosen, for some specific iterations, to accept the step computed by our method without giving attention to the function value. As we will see later, this choice allows us to maintain a superlinear convergence result.

Notice that for any $x \in U(x)$, we have that $f$ behaves smoothly along $s$ at $x$, since the $s$-directional derivatives of $\phi_i$ are all the same for $i \in I(x)$. 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 $Z_x^\triangledown$ 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 $r \in \{1, \ldots, n-1\}$. The cases $r = 0$ and $r = n$ will be treated later (see Remark 3).

In light of Remark 2, due to Assumption 2, it is possible to see that the map $J_x : \mathbb{R}^n \rightarrow \mathbb{R}^r$ is surjective for all $x$ in a small neighborhood $\mathcal{N}$ of $x^*$. Hence, for $x \in \mathcal{N}$, there must exist $J_x^\triangledown \in \mathbb{R}^{n \times r}$ such that $J_x J_x^\triangledown = I_r$. Moreover, by [2, Lemma 14.3], one can see that there is only one map $Z_x : \mathbb{R}^n \rightarrow \mathbb{R}^{(n-r) \times n}$ $x \mapsto Z_x$ such that $Z_x J_x^\triangledown$ is a null matrix, $Z_x Z_x^\triangledown = I_{n-r}$ and the following relations hold

$$
Z_x^\triangledown Z_x + J_x^\triangledown J_x = I_n \text{ and } J_x Z_x^\triangledown = 0.
$$

So, we may divide $\mathbb{R}^n$ into two subspaces, generated by the columns of $Z_x^\triangledown$ and $J_x^\triangledown$, respectively.
Now, coming back to the optimization problem (23), we define its Lagrangian function

\[ L(x, \lambda) = \phi_{r+1}(x) + \lambda^T \Phi(x). \]  

By Remark 2, the feasible set of problem (23) satisfies the linear independence constraint qualification and thus there is only one \( \lambda_* \in \mathbb{R}^r \) such that \( \nabla_x L(x_*, \lambda_*) \) is the null vector. So, in possession of this vector \( \lambda_* \), we define

\[ g(x) := Z_x^a \nabla_x L(x, \lambda_*) \overset{(24)}{=} Z_x^a \nabla \phi_{r+1}(x). \]  

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

\[ A_k := I_n - Z_{x_k}^a \hat{H}_k^{-1} Z_{x_k}^a \trans{H}_k, \]  

with \( \hat{H}_k := Z_{x_k}^a \trans{H}_k Z_{x_k}^a \).

Below, we present a theorem that establishes the exact solution \( d_{k, t_k} \), obtained in (5), whenever it is equivalent to (22). For this result and the subsequent ones, we define

\[ \tau_{k, \bar{t}_k} := \max_{1 \leq i \leq r+1} \left\| x_{k,i} - x_k \right\|. \]  

**Theorem 2** Under Assumptions 1, 2 and 3, suppose we are at a fixed outer iteration \( k \) of GraFuS, and at the last inner iteration indexed by \( \bar{t}_k \). Then, if \( k \in K \), where \( K \) is the index set whose existence was established in Corollary 2, and \( x_k \in N \), where \( N \) is the small neighborhood in which the map \( J_x \) is surjective, we have that \( d_{k, \bar{t}_k} = d_{k, \bar{t}_k}^{U} + d_{k, \bar{t}_k}^{V} \),

with

\[ d_{k, \bar{t}_k}^{U} := -Z_{x_k}^a \hat{H}_k^{-1} g(x_k) + \rho_{k}^{U} \]  

and

\[ d_{k, \bar{t}_k}^{V} := -A_k J_{x_k}^a \Phi(x_k) + \rho_{k}^{V}, \]  

where

\[ \rho_{k}^{U} = -Z_{x_k}^a \hat{H}_k^{-1} Z_{x_k}^a \trans{H}_k \bar{\rho}_k \]  

and

\[ \rho_{k}^{V} = -A_k J_{x_k}^a \hat{\rho}_k, \]  

for some \( \bar{\rho}_k \in \mathbb{R}^n \) and \( \hat{\rho}_k \in \mathbb{R}^r \) satisfying

\[ \| \bar{\rho}_k \| = O \left( \tau_{k, \bar{t}_k} \right) \]  

and

\[ \| \hat{\rho}_k \| = O \left( \tau_{k, \bar{t}_k}^2 \right) + O \left( \tau_{k, \bar{t}_k} \right) O \left( \| d_{k, \bar{t}_k} \| \right), \]  

with \( \tau_{k, \bar{t}_k} \) defined in (28).

**Proof** First, we consider the Karush–Kuhn–Tucker conditions of problem (22), which tell us that the solution \( d_{k, \bar{t}_k} \) must satisfy

\[ \tilde{\phi}_k + \bar{J}_k d_{k, \bar{t}_k} = 0 \]  

\( \oplus \) Springer
and
\[
\nabla \phi_{r+1}(x_{k,r+1}) + H_k d_{k,1_k} + \tilde{J}_k^T \tilde{\lambda} = 0,
\]
(30)
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 (29) and (30) that
\[
0 = \Phi(x_k) + J_{x_k} d_{k,1_k} + \left[ \Phi_k - \Phi(x_k) \right] + \left[ \tilde{J}_k - J_{x_k} \right] d_{k,1_k}
\]
(31)
and
\[
\nabla \phi_{r+1}(x_k) + H_k d_{k,1_k} + J_{x_k}^T \tilde{\lambda} + \tilde{\rho}_k = 0,
\]
(32)
where $\|\tilde{\rho}_k\| = O\left(\tau_{k,1_k}^2\right) + O\left(\tau_{k,1_k}\right) O\left(\|d_{k,1_k}\|\right)$ and $\|\tilde{\rho}_k\| = O\left(\tau_{k,1_k}\right)$. Then, because $A_k J_{x_k}^a$ is a right inverse for $J_{x_k}$ (see [2, Section 14.2] or simply use the fact that $J_{x_k} Z_{x_k}^a = 0$), it is possible to decompose $\mathbb{R}^n$ in two subspaces generated by the columns of $Z_{x_k}^a$ and $A_k J_{x_k}^a$. As a consequence, we can consider two vectors $d_{k,U}^a$ and $d_{k,V}^a$ such that there exist $\alpha_U$ and $\alpha_V$ that imply $d_{k,1_k} = d_{k,U}^a + d_{k,V}^a$, with $d_{k,U}^a = Z_{x_k}^a \alpha_U$ and $d_{k,V}^a = A_k J_{x_k}^a \alpha_V$. Hence, looking at relation (31), we obtain that $\alpha_V = -\Phi(x_k) - \tilde{\rho}_k$, which yields $d_{k,V}^a = -A_k J_{x_k}^a \Phi(x_k) + \rho_k^V$, with $\rho_k^V = -A_k J_{x_k}^a \tilde{\rho}_k$.

Finally, pre-multiplying relation (32) by $Z_{x_k}^a T$, we have
\[
g(x_k) + Z_{x_k}^a H_k \left[ Z_{x_k}^a \alpha_U - A_k J_{x_k}^a \left( \Phi(x_k) + \hat{\rho}_k \right) \right] + Z_{x_k}^a T \tilde{\rho}_k = 0.
\]
Then, since $Z_{x_k}^a H_k A_k = 0$, we complete the proof by noticing that
\[
\alpha_U = -\hat{H}_k^{-1} g(x_k) - \hat{H}_k^{-1} Z_{x_k}^a T \tilde{\rho}_k \Rightarrow d_{k,U}^a = -Z_{x_k}^a \hat{H}_k^{-1} g(x_k) + \rho_k^U,
\]
where $\rho_k^U = -Z_{x_k}^a \hat{H}_k^{-1} Z_{x_k}^a T \tilde{\rho}_k$. \hfill \Box

Below, we present the last technical result before providing the key theorem of this subsection. As an hypothesis of this statement, we assume that, for $k \in \mathcal{K}$, the matrices $H_k$ must converge to a matrix $H_*$, where
\[
H_* = \nabla^2_{x_*} \mathcal{L}(x_*, \lambda_*) + \gamma J_{x_*}^T J_{x_*}, \quad \text{for some } \gamma \geq 0.
\]
(33)
In other words, we are saying that there must exist $\gamma > 0$ such that the difference between the matrix $H_k$ and $\nabla^2_{x_*} \mathcal{L}(x_*, \lambda_*)$ tends to $\gamma J_{x_*}^T J_{x_*}$, when $k \in \mathcal{K}$. Although this is stronger than the Dennis-Moré like condition [47, Theorem 18.5], assumptions asking for convergence to some target matrix $H_*$ can be found [2, Theorem 14.7].

\[\square\] Springer
By Assumption 3, we see that the Hessian of the Lagrangian must be a positive definite matrix with respect to the subspace $U(x_\ast)$. So, for $\gamma > 0$ sufficiently large, $H_\ast$ also becomes a positive definite matrix.

**Theorem 3** Under Assumptions 1, 2 and 3, suppose that $\{x_k\}$ is an infinite and bounded sequence generated by GraFuS with $\nu_{\text{opt}} = 0$. Assume that $k \in K$, where $K$ is the index set defined in Corollary 2, and $x_k \in N$, where $N$ is the small neighborhood in which the map $J_x$ is surjective. Also, close to $x_\ast$, suppose that the maps $Z^\varphi_x : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times (n-r)}$ and $J^\varphi_x : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times r}$ are all Lipschitz continuous functions and that the reduced gradient given in (26) satisfies $g \in C^1$ with $g'$ being also a Lipschitz continuous function close to $x_\ast$. Moreover, assume that $H_k \rightarrow H_\ast$ with $H_\ast$ being the matrix presented in (33). Additionally, suppose that, close to $x_\ast$, we have $\|H_k - H_\ast\| = O(\|x_k - x_\ast\|)$, for $k \in K$. Then, the following relation holds

$$\|x_{k+1} - x_\ast\| = O(\|x_k - x_\ast\|^2) + \rho_k^U + \rho_k^V,$$

for $k \in K$, with $\rho_k^U$ and $\rho_k^V$ from Theorem 2.

**Proof** First, let us define $\tilde{x}_{k+1} := x_k + A_{x_\ast}^T H^{-1}_\ast Z^\varphi_{x_\ast}^T H_\ast$, with $H_\ast = Z^\varphi_{x_\ast}^T H_\ast Z^\varphi_{x_\ast}$. Using this fact, considering the Taylor expansion of the map $\Phi$ around $x_\ast$ and remembering that $\Phi(x_\ast) = 0$ in the equality $(\ast)$ below and noticing that $J^\varphi_x$ is Lipschitz continuous and a bounded map around $x_\ast$ in $(\ast \ast)$, we have, for a sufficiently small neighborhood of $x_\ast$, that

$$\tilde{x}_{k+1} - x_\ast = x_k - x_\ast - A_k J^\varphi_{x_k} \Phi(x_k) + \rho_k^V \equiv (\ast) x_k - x_\ast - A_k J^\varphi_{x_k} J_{x_\ast}(x_k - x_\ast) + O(\|x_k - x_\ast\|^2) + \rho_k^V \equiv (\ast \ast) x_k - x_\ast - A_k J^\varphi_{x_k} J_{x_\ast}(x_k - x_\ast) + O(\|x_k - x_\ast\|^2) + \rho_k^V .$$

Consequently, taking into account the relation $g'(x_\ast) = Z^\varphi_{x_\ast}^T \nabla^2_{xx} L(x_\ast, \lambda_\ast)$ (see [2, Section 14.5]) in $(\bullet)$, the Lipschitz property around $x_\ast$ of the maps $Z^\varphi$ and $H^{-1}$ in
\[ x_{k+1} - x_* = x_k + d^V_{k,\tilde{\ell}_k} + d^U_{k,\tilde{\ell}_k} - x_* \]
\[ = \tilde{x}_{k+1} - x_* - Z^x_{k} \hat{H}^{-1}_k g(x_k) + \rho^U_k \quad (\text{recall that } \tilde{x}_{k+1} := x_k + d^V_{k,\tilde{\ell}_k}) \]
\[ \equiv \tilde{x}_{k+1} - x_* - Z^x_{k} \hat{H}^{-1}_k Z^x_{\ell} \nabla^2_{xx} L(x_*, \lambda_*)(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho^U_k \]
\[ \equiv \tilde{x}_{k+1} - x_* - Z^x_{k} \hat{H}^{-1}_k Z^x_{\ell} \nabla^2_{xx} L(x_*, \lambda_*)(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho^U_k + \rho^V_k \]
\[ \equiv A_* Z^x_{k} Z_{\ell}(x_k - x_*) + O(\|x_k - x_*\|^2) + \rho^U_k + \rho^V_k. \]

Hence, since \( A_* Z^x_{k} = 0 \), it yields \( \|x_{k+1} - x_*\| = O(\|x_k - x_*\|^2) + \rho^U_k + \rho^V_k \), which concludes the proof. \( \square \)

In the above result, we require that \( \|H_k - H_*\| = O(\|x_k - x_*\|) \) for any \( k \in K \) and \( x_k \) close enough to \( x_* \). This was done in order to obtain the best possible result, i.e., \( \|x_{k+1} - x_*\| = O(\|x_k - x_*\|^2) + \rho^U_k + \rho^V_k \). Notice that, if we had asked for \( H_k \rightarrow H_* \) only, we would have obtained just \( \|x_{k+1} - x_*\| = o(\|x_k - x_*\|) + \rho^U_k + \rho^V_k \), which does not invalidate the next theorem.

Finally, we are able to prove the most important result of this manuscript, which ensures that, under special circumstances, the method either moves superlinearly to a minimizer of the problem or superlinearly reduces the optimality certificate.

**Theorem 4** Under the conditions of Theorem 3, if \( \sigma_k > 1 \) for all \( k \in K \), we have
\[
\min \left\{ \frac{v_{k+1}}{v_k}, \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \right\} \rightarrow 0.
\]

**Proof** Suppose, by contradiction, that there exist an infinite index set \( \hat{K} \subset K \) and \( M > 0 \) such that
\[
\min \left\{ \frac{v_{k+1}}{v_k}, \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \right\} > M, \text{ for all } k \in \hat{K}. \tag{34}
\]

Therefore, \( v_{k+1}/v_k > M \), for all \( k \in \hat{K} \), which yields, by the way the algorithm was designed, that \( \|H_k^{-1} G_{k,\tilde{\ell}_k} \lambda_{k,\tilde{\ell}_k} \| \geq M v_k \), for all \( k \in \hat{K} \). Now, since \( \epsilon_{k,\tilde{\ell}_k} = O(v_k) \) and \( d_{k,\tilde{\ell}_k} = -H_k^{-1} G_{k,\tilde{\ell}_k} \lambda_{k,\tilde{\ell}_k} \) is a valid relation for the primal-dual variables that solve
the quadratic programming problem that appears in Step 2 (when the trust-region constraints are not active), we have that \( \| d_{k, \hat{J}_k} \| \to 0 \) (because of the last part in the proof of Lemma 6) and

\[
\epsilon_{k, \hat{J}_k} = O(\| d_{k, \hat{J}_k} \|) = O(\| x_{k+1} - x_k \|), \quad \text{for all } k \in \hat{K}.
\]

Since \( \| d_{k, \hat{J}_k} \| \to 0 \) and we assume \( x_k \to x_* \), it follows that \( x_{k+1} \to x_* \). Moreover, because (34) is assumed, we have

\[
\frac{\| x_{k+1} - x_k \|}{\| x_{k+1} - x_* \|} \leq 1 + \frac{\| x_k - x_* \|}{\| x_{k+1} - x_* \|} < 1 + \frac{1}{M} = \frac{M + 1}{M},
\]

which assures that \( \epsilon_{k, \hat{J}_k} = O(\| x_{k+1} - x_* \|) \) and \( d_{k, \hat{J}_k} = O(\| x_{k+1} - x_* \|) \). Consequently, since \( \sigma_k > 1 \), it yields that

\[
\tau_{k, \hat{J}_k} = O \left( \left( \epsilon_{k, \hat{J}_k} \right)^{\sigma_k} \right) = o(\| x_{k+1} - x_* \|),
\]

and, by Theorem 2, we see that \( |\hat{\rho}_k| = o(\| x_{k+1} - x_* \|) \) and \( |\bar{\rho}_k| = o(\| x_{k+1} - x_* \|) \), which ensures, by Theorem 3, that

\[
\| x_{k+1} - x_* \| = O(\| x_k - x_* \|^2) + o(\| x_{k+1} - x_* \|), \quad \text{for all } k \in \hat{K}.
\]

So, for all \( k \in \hat{K} \) sufficiently large, \( \| x_{k+1} - x_* \| = O(\| x_k - x_* \|) \). However, the above relation contradicts the initial assumption (34). Therefore, we must have

\[
\min \left\{ \frac{v_{k+1}}{v_k}, \frac{\| x_{k+1} - x_* \|}{\| x_k - x_* \|} \right\} \to 0, \quad \text{for all } k \in \hat{K}.
\]

\[\Box\]

**Remark 3** The local convergence results were developed assuming \( r \in \{1, \ldots, n-1\} \). For the case \( r = 0 \), we have that the method is approaching a point for which the function \( f \) is smooth in the whole neighborhood. For such a situation, it is straightforward to see that the direction \( d_{k, \hat{J}_k} \) will have only the \( \mathcal{U} \)-component, i.e., \( d_{k, \hat{J}_k} = d^{\mathcal{U}}_{k, \hat{J}_k} \) with \( Z^x = I_n \) 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 that the direction \( d_{k, \hat{J}_k} \) will have only the \( \mathcal{V} \)-component, i.e., \( d_{k, \hat{J}_k} = d^{\mathcal{V}}_{k, \hat{J}_k} \) with \( A_k = I_n \) for all \( x_k \) around \( x_* \). Therefore, in both cases, the result of Theorem 4 will be preserved, but, for the case that \( r = n \), the value \( \sigma_k \) does not need to be strictly greater than one, i.e., in such a case Theorem 4 holds for \( \sigma_k = 1 \).

### 5 Numerical results

This section has the intent to illustrate the main local convergence results obtained. 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 provide the reader with proof-of-concept numerical 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.

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 [31]) and (ii) the GraFuS method. We have used the original GS implementation without any modification (with the exception of using a nonnormalized search direction).\(^2\) For completeness, we present the parameter values used in Algorithm 1: $m = 2n$; $v_0 = 10^{-6}$; $\epsilon_0 = 10^{-1}$; $\nu_{\text{opt}} = 10^{-6}$; $\epsilon_{\text{opt}} = 10^{-6}$; $\theta_v = 1$; $\theta_\epsilon = 10^{-1}$; $\gamma = 0.5$ and $\beta = 0$.

The parameter values used in GraFuS were: $m = 2n$; $v_0 = 10^{-2}$; $\nu_{\text{opt}} = 10^{-6}$; $\gamma_{\epsilon} = 4$; $\gamma_{\Delta} = 4$; $\delta = 0.90$; $\varrho = 1.50$; $\rho = 10^{-8}$ and $\theta = 0.5$. The value of $\sigma_k$ in Step 1 was set as follows. We start the algorithm with $\sigma_0 = 1$ and, setting $|\lambda|_\#$ as the number of entries of $\lambda$ greater than $10^{-3}/(n + 1)$, we have updated $\sigma_k$ every time a reduction on $\nu_k$ was performed in such a way that

$$\sigma_{k+1} = \begin{cases} 1, & |\lambda_k, T_k|_\# \geq n + 1 \\ 1.5, & \text{otherwise} \end{cases}$$

Notice that $|\lambda|_\# - 1$ tries to approximate the dimension of the subspace $\mathcal{V}(x_\ast)$.

An important aspect that we must recall 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 bootstrap 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.

We also must stress that although the optimality certificates of Algorithms 1 and 2 are very similar, they are not the same (specially because the quadratic programming problem of each method is different). Therefore, one might be more rigorous than the other one. Thus, although in most problems the GraFuS method appears to be closer to the solution, this does not mean that GS is not able to reach the same precision (maybe a tighter optimality parameter would allow it).

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

\(^2\) The GS code can be found at http://cs.nyu.edu/overton/papers/gradsamp/alg/.
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 somehow approach a matrix of the form presented in (33). 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 [19] for the optimization problem that appears in (23).

5.1 $H_k$ updates in GraFuS method

As we have seen in the last section, if $k \in K$ is large enough, it is possible to see the quadratic programming problem that is solved in the (outer, inner) iteration $(k, l_k)$ 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 (25). 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

$$H_k(x_k + - x_k - ) = \nabla x L(x_k + \lambda_+) - \nabla x L(x_k - \lambda_-),$$

where $L$ is the Lagrangian function defined in (25) and $\lambda_+$ and $\lambda_-$ are vectors that try to approximate the multiplier $\lambda_*$ that fulfills (26). In addition,

$$\nabla x L(x, \lambda) = \nabla \phi_{r+1}(x) + \sum_{i=1}^r \lambda_i (\nabla \phi_i(x) - \nabla \phi_{r+1}(x))$$

$$= \left(1 - \sum_{i=1}^r \lambda_i\right) \nabla \phi_{r+1}(x) + \sum_{i=1}^r \lambda_i \nabla \phi_i(x).$$

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 $\nabla x L(x, \lambda) = \hat{G} \hat{\lambda}$, where

$$\hat{G} := [\nabla \phi_1(x) \ldots \nabla \phi_{r+1}(x)].$$

Hence, if in two fixed pairs $(k_+, l_+)$ and $(k_-, l_-)$ of (outer, inner) iterations we have good sets of sampled points (in the sense that the conditions $i$ and $ii$ of Lemma 6 are valid), 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 a good set of sampled points. In fact, we could say that all iterations in $\tilde{K} = \{k \in \mathbb{N} : \nu_{k+1} < \nu_k\}$ must produce a good set of sampled points, but to restrain the update of $H_k$ just for those iterations can lead us to very few updates during the execution of the method. So, 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 pairs $(k_+, l_+)$ and $(k_-, l_-)$ of (outer, inner) iterations such that

$$
\|G_{k_+, l_+} \lambda_{k_+, l_+}\| \leq \sqrt{\nu_{k_+}} \quad \text{and} \quad \|G_{k_-, l_-} \lambda_{k_-, l_-}\| \leq \sqrt{\nu_{k_-}}.
$$

Set $x_+ := x_{k_+}, x_- := x_{k_-}, v_+ := G_{k_+, l_+} \lambda_{k_+, l_+}$ and $v_- := G_{k_-, l_-} \lambda_{k_-, l_-}$.

**Step 1** Set $p := x_+ - x_- \quad \text{and} \quad q := v_+ - v_-$. 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 $\hat{k}$ and an inner iteration $\hat{l}$ such that $\|H_{\hat{k}}^{-1} G_{k, l} \hat{\lambda}_{k, l}\| \leq \sqrt{\nu_k}$. Then, $x_- \leftarrow x_+, x_+ \leftarrow x_{\hat{k}}, v_- \leftarrow v_+, v_+ \leftarrow G_{\hat{k}, l} \hat{\lambda}_{\hat{k}, l}$. Go back to Step 1.

Clearly, other ways of updating $H_k$ are possible. Indeed, even the pure BFGS update as considered in [36] can be performed (although, in such a case, we have to assume that for all iterates the function $f$ will be differentiable and Assumption 1 will no longer be satisfied). We believe that an improvement on the updating of $H_k$ may be an important advance on the performance of GraFuS.

### 5.2 Illustrative examples

The functions that were solved to illustrate our algorithm are the following [23]:

**F1** Chained CB3 I

$$
\begin{align*}
    f(x) &= \sum_{i=1}^{n-1} \max \left\{ x_i^4 + x_{i+1}^2, (2 - x_i)^2 + (2 - x_{i+1})^2, 2 \exp(-x_i + x_{i+1}) \right\};
\end{align*}
$$

**F2** Chained CB3 II

$$
\begin{align*}
    f(x) &= \max \left\{ \sum_{i=1}^{n-1} \left( x_i^4 + x_{i+1}^2 \right), \sum_{i=1}^{n-1} \left( (2 - x_i)^2 + (2 - x_{i+1})^2 \right), \sum_{i=1}^{n-1} 2 \exp(-x_i + x_{i+1}) \right\};
\end{align*}
$$

© Springer
(F3) Nonsmooth generalization of Brown function 2

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

(F4) Chained crescent I

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

The first two functions are convex, whereas the last two ones are nonconvex functions. In addition, F1 and F3 satisfy \( U(x_*) = \{0\} \), a condition that does not hold for F2 and F4.

To observe the GraFuS functioning and to put it into perspective with GS, we have comparatively examined the CPU time and the number of iterations versus \( f(x_k) - f_* \), where \( f_* \) is set as the best function value obtained by the methods in all of the runs. Because of the nondeterministic nature of the methods, we have used the median and quartiles (25% and 75%) of the twenty runs. As a complementary tool for assessing how fast our method goes towards the optimal function value, in the plots with the number of iterations, we have represented the value

\[ \min \left\{ \frac{f(x_{k+1}) - f_*}{f(x_k) - f_*}, 1 \right\} \]

with color scales along the plotted curves of GraFuS, where a brighter hue stands for a value close to zero, and a darker color for the values near one. Notice that the values above 1 must have a safeguard, because since our method might not be monotone, the ratio \( (f(x_{k+1}) - f_*)/(f(x_k) - f_*) \) could be greater than one. The reader will see that a few increases on the function value appears in the figures that are shown below. This is due to the fact that the measure (quartiles) used to represent the twenty runs somehow absorbs the nonmonotone behavior of the function values.

Additionally, we have examined the values \( v_{k+1}/v_k \) and \( \|x_{k+1} - x_*\|/\|x_k - x_*\| \) for \( k \) such that \( v_{k+1} < v_k \). For this two measures, a detailed explanation must be given on how we have plotted the corresponding curves. As we have mentioned before, we have solved each function more than once. However, an iteration \( k \) for which \( v_{k+1} < v_k \) occurs is not necessarily the same iteration where a second run will have \( v_{k+1} = v_k \). It is only possible to track these values for different runs if instead of looking at the iteration \( k \), we monitor the actual occurrences of \( v_{k+1} < v_k \). Therefore, we have proceeded in the following way. For each run, we set the \( w \)-dimensional vectors

\[ \text{vec}_v \leftarrow \left[ \frac{v_{k_1+1}}{v_{k_1}}, \ldots, \frac{v_{k_w+1}}{v_{k_w}} \right] \] and \( \text{vec}_{x_*} \leftarrow \left[ \frac{\|x_{k_1+1} - x_*\|}{\|x_{k_1} - x_*\|}, \ldots, \frac{\|x_{k_w+1} - x_*\|}{\|x_{k_w} - x_*\|} \right], \]
where $k_i$ is the iteration that for the $i$-th time, $\nu_{k_i+1} < \nu_{k_i}$ has occurred. Moreover, for the case that $w < 30$, we enlarge the vectors $\text{vec}_\nu$ and $\text{vec}_{x^*}$ by copying the last value of each vector, respectively, until it reaches 30 dimensions. This is necessary because not every run of GraFuS will give vectors with equal dimensions. Then, the quartiles are computed using the vectors $\text{vec}_\nu$ and $\text{vec}_{x^*}$ of each run.

In Figs. 1 and 2, we see the results obtained by the runs related to the first function $F_1$. It is possible to observe that GraFuS has a good performance in all the measures. Not only a high precision is achieved, but one can also see

$$\min \left\{ \frac{\nu_{k+1}}{\nu_k}, \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \right\}$$

approaching zero, in accordance with the result of Theorem 4. On the other hand, although Fig. 3 presents good results for the function $F_2$, the sequence that appears in Theorem 4 does not approach zero as fast as it happens for $F_1$ (see Fig. 4). This has a reasonable explanation. Notice that the result of Theorem 4 is conditioned by Theorem 3, which has, as an assumption, that the matrices $H_k$, with $k \in \mathcal{K}$, must
A fast gradient and function sampling method for…

\[ x^* = e, \quad a = 5, \quad b = 10, \]

Fig. 2 The simple black line plot and the one with diamond marks represent, respectively, the medians of the vectors \( \text{vec}_k \) and \( \text{vec}_{x^*_k} \) for the function \( F_1 \). For both \( n = 5 \) and \( n = 10 \), we have \( x^*_k = e, \quad a = 5, \quad b = 10, \) approach \( H_k \) satisfying (33). However, since \( F_1 \) has \( U(x^*_k) = \{0\} \), the matrices \( H_k \) do not need to contain any kind of second-order information to guarantee Theorem 4 to hold, which is the reason why (35) approaches quickly to zero when GraFuS is applied to this function. However, \( F_2 \) has \( \dim U(x^*_k) = n - 2 \), which means that the result of Theorem 4 will be conditioned to how good is the approximation of \( H_k \) to \( H^*_k \) at each iteration \( k \in K \).

The function \( F_3 \) has some interesting features, since it does not admit a representation as a maximum of a finite number of smooth functions. Indeed, let us consider the function \( h(a, b) = a^{(1+b^2)} \), for \( a \geq 0 \). Then, it yields that

\[
\lim_{\varepsilon \downarrow 0} \frac{\partial h}{\partial a}(\varepsilon, \varepsilon) = \lim_{\varepsilon \downarrow 0} (1 + \varepsilon^2)e^{\varepsilon^2} = 1;
\]

\[
\lim_{\varepsilon \downarrow 0} \frac{\partial h}{\partial a}(2^{-1/\varepsilon^3}, \varepsilon) = \lim_{\varepsilon \downarrow 0} (1 + \varepsilon^2)2^{-1/\varepsilon} = 0.
\]

So, it is possible to see that any representation of \( F_3 \) that might involve a maximum of functions cannot have smooth functions. Therefore, this function does not satisfy the requirements of our convergence analysis. However, this does not prevent GraFuS to have a good performance (see Figs. 5, 6).

Looking at the results obtained for the function \( F_4 \) in Figs. 7 and 8, the analysis follows very closely the one that was presented for function \( F_2 \). Since \( F_4 \) satisfies \( U(x^*_k) = n - 1 \), as depicted in Fig. 8, the value (35) does not go to zero as quickly as it could be expected. Nevertheless, the results obtained in Fig. 7 also show a good behavior of GraFuS.

Finally, we stress that we have experienced good and bad results when GraFuS is applied to problems with higher dimension. For the nonconvex functions considered in this study, the proposed method keeps presenting a good behavior, whereas for the convex optimization problems, the GS method shows better numerical results.
Fig. 3 Medians and quartiles of twenty runs of GS and GraFuS methods for the function $F_2$. The black line plots represent the GS method, whereas the grey/colored continuous ones with diamond marks stand for GraFuS. For both $n = 5$ and $n = 10$, we have $x^* = e$. a $n = 5$, b $n = 5$, c $n = 10$, d $n = 10$ (Color figure online)

Fig. 4 The simple black line plot and the one with diamond marks represent, respectively, the medians of the vectors $\text{vec}_v$ and $\text{vec}_{x^*}$ for the function $F_2$. For both $n = 5$ and $n = 10$, we have $x^* = e$. a $n = 5$, b $n = 10$
Fig. 5 Medians and quartiles of twenty runs of GS and GraFuS methods for the function $F_3$. The black line plots represent the GS method, whereas the grey/colored continuous ones with diamond marks stand for GraFuS. For both $n = 5$ and $n = 10$, we have $x^* = 0$. a $n = 5$, b $n = 5$, c $n = 10$, d $n = 10$ (Color figure online)

Fig. 6 The simple black line plot and the one with diamond marks represent, respectively, the medians of the vectors $\text{vec}_y$ and $\text{vec}_{x^*}$ for the function $F_3$. For both $n = 5$ and $n = 10$, we have $x^* = 0$. a $n = 5$, b $n = 10$
Fig. 7 Medians and quartiles of twenty runs of GS and GraFuS methods for the function $F_4$. The black line plots represent the GS method, whereas the grey/colored continuous ones with diamond marks stand for GraFuS. For both $n = 5$ and $n = 10$, we have $x_\ast = 0$. a $n = 5$, b $n = 5$, c $n = 10$, d $n = 10$ (Color figure online)

Fig. 8 The simple black line plot and the one with diamond marks represent, respectively, the medians of the vectors $\text{vec}_v$ and $\text{vec}_{x_\ast}$ for the function $F_4$. For both $n = 5$ and $n = 10$, we have $x_\ast = 0$. a $n = 5$, b $n = 10$
6 Conclusions

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 for nonconvex nonsmooth unconstrained minimization.

Additionally, we believe that an important step has been taken in the direction of obtaining a rapid method for minimizing 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 illustrative examples considered in the numerical experiments, one can see that fast moves are not rare and can be expected for a reasonable amount of iterations. However, it must be stressed that the iterations of GraFuS are computationally expensive when compared to GS, and for this reason, the rapid behavior of GraFuS might not be translated to a faster method for some functions.

The matters of efficiency and applicability of the method have not been 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 provide proof-of-concept numerical instances that corroborate the main theoretical results. There are many possibilities of improvements on the algorithm, e.g. different forms of updating the matrices $H_k$, efficient ways of selecting the sampled points without affecting the global convergence, and, from a more practical standpoint, the possibility of taking advantage of the gradient computation (whenever this can be done) to obtain the function values at the sampled points.

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

– under which conditions could we establish $H_k \to H_*$?
– can one ensure the global convergence of GraFuS without the assumption that $f$ must have bounded level sets?
– would it be possible to have convergence results with more general assumptions, e.g. without supposing any minimax structure for the optimization problem and/or without Assumption 2?

Acknowledgements We are greatly thankful to the anonymous referees who carefully read a previous version of this paper and contributed with valuable suggestions to improve our manuscript.

Appendix

The aim of this appendix is to show that the assumption $(\lambda_k, I_k)_i = 0$, whenever $i \notin \mathcal{I}(x_*)$ and $k \in \mathcal{K}$, with $\mathcal{K}$ defined in Corollary 2, is not necessary. For this goal, we will show that even without such an assumption, the results from the local convergence subsection remain the same.
We divide our reasoning in two cases and remind the reader that we have assumed \( \mathcal{I}(x_\ast) = \{1, \ldots, r + 1\} \):

(A1) The cardinality of \( \mathcal{I}(x_\ast) \) is \( n + 1 \);
(A2) The cardinality of \( \mathcal{I}(x_\ast) \) is \( r + 1 \) with \( r < n \).

Suppose first that A1 holds and let us consider an iterate \( x_k \) sufficiently close to \( x_\ast \). Moreover, assume that \( k \in \mathcal{K} \), where \( \mathcal{K} \) is the index set defined in Corollary 2. Then, looking at the optimization problem in (22), we see that any additional active constraint will generate an additional active constraint to (22) in a way that it will be a linear combination of the first \( n + 1 \) active constraints (by Remark 2 and because the rank of \( \tilde{J}_k \) remains constant in a close neighborhood of \( x_\ast \)). 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

\[
\min_{(d,z) \in \mathbb{R}^{n+1}} \ z + \frac{1}{2} d^T \mathbf{H}_k d \\
\text{s.t. } f \left( \mathbf{x}^j_k \right) + \nabla f \left( \mathbf{x}^j_k \right)^T \left( \mathbf{x}_k + d - \mathbf{x}^j_k \right) = z, \quad 1 \leq i \leq r + 2,
\]

where here we assume that rearrangements were done in order to have the additional constraint as the \( (r + 2) \)-th constraint and that it has the associated sampled point \( x_{k,r+2}^j \). 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 \( \phi_i \), that only the functions \( \phi_1, \ldots, \phi_{r+1} \) can assume the maximum at any sampled point. So, there exists \( j \in \{1, \ldots, r + 1\} \) such that \( f(\mathbf{x}_{k,r+2}^j) = \phi_j(\mathbf{x}_{k,r+2}^j) \). Consequently, recalling that \( k \in \mathcal{K} \), the dual problem of the above minimization problem can be seen as

\[
\max_{\lambda \in \mathbb{R}^{r+2}} \sum_{i=1}^{r+1} \lambda_i \left[ \phi_i \left( \mathbf{x}^j_k \right) + \nabla \phi_i \left( \mathbf{x}^j_k \right)^T \left( \mathbf{x}_k - \mathbf{x}^j_k \right) \right] \\
+ \lambda_{r+2} \left[ \phi_j \left( \mathbf{x}^j_{k,r+2} \right) + \nabla \phi_j \left( \mathbf{x}^j_{k,r+2} \right)^T \left( \mathbf{x}_k - \mathbf{x}^j_{k,r+2} \right) \right] \\
- \frac{1}{2} \left[ \sum_{i=1}^{r+1} \lambda_i \nabla \phi_i \left( \mathbf{x}^j_k \right) + \lambda_{r+2} \nabla \phi_j \left( \mathbf{x}^j_{k,r+2} \right) \right]^2_{\mathbf{H}_k^{-1}} \\
\text{s.t. } e^T \lambda = 1.
\]
Therefore, we can turn this last constrained maximization problem into an unconstrained one by making the following substitution $\lambda_{r+2} = 1 - \sum_{i=1}^{r+1} \lambda_i$. So, we have

$$
\max_{\lambda \in \mathbb{R}^{r+1}} \sum_{i=1}^{r+1} \lambda_i \left[ \phi_i \left( x_{k,i}^T \right) + \nabla \phi_i \left( x_{k,i}^T \right)^T \left( x_k - x_{k,i}^T \right) - \phi_j \left( x_{k,r+2}^T \right) \right]
$$

$$
- \nabla \phi_j \left( x_{k,r+2}^T \right)^T \left( x_k - x_{k,r+2}^T \right) + \nabla \phi_j \left( x_{k,r+2}^T \right)
$$

$$
+ \nabla \phi_j \left( x_{k,r+2}^T \right)^T \left( x_k - x_{k,r+2}^T \right)
$$

$$
- \frac{1}{2} \sum_{i=1}^{r+1} \lambda_i \left[ \nabla \phi_i \left( x_{k,i}^T \right) - \nabla \phi_j \left( x_{k,r+2}^T \right) \right] + \nabla \phi_j \left( x_{k,r+2}^T \right) ^2 \right) H_k^{-1}.
$$

Since the above problem is concave and smooth, its solution $\bar{\lambda} \in \mathbb{R}^{r+1}$ can be obtained by equaling the derivative of the objective function to the null vector. Consequently, assuming without loss of generality that the function $\phi_j$ involved in the additional constraint is $\phi_{r+1}$, we have

$$
\left( \nabla \phi_1 \left( x_{k,1}^T \right) - \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \right) \vdots \left( \nabla \phi_1 \left( x_{k,r+1}^T \right) - \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \right)
$$

$$
H_k^{-1} \left( \nabla \phi_1 \left( x_{k,1}^T \right) - \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \right) H_k^{-1} \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \left( \nabla \phi_1 \left( x_{k,1}^T \right) - \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \right)
$$

$$
= \left( \phi_1 \left( x_{k,1}^T \right) + \nabla \phi_1 \left( x_{k,1}^T \right) \left( x_k - x_{k,1}^T \right) \right) \vdots \left( \phi_{r+1} \left( x_{k,r+1}^T \right) + \nabla \phi_{r+1} \left( x_{k,r+1}^T \right) \left( x_k - x_{k,r+1}^T \right) \right)
$$

$$
- \left( \phi_{r+1} \left( x_{k,r+2}^T \right) + \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \left( x_k - x_{k,r+2}^T \right) \right) \vdots \left( \phi_{r+1} \left( x_{k,r+2}^T \right) + \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \left( x_k - x_{k,r+2}^T \right) \right)
$$

$$
- \left( \nabla \phi_1 \left( x_{k,1}^T \right) - \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \right) \vdots \left( \nabla \phi_1 \left( x_{k,1}^T \right) - \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \right) \right) H_k^{-1} \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \left( \nabla \phi_1 \left( x_{k,1}^T \right) - \nabla \phi_{r+1} \left( x_{k,r+2}^T \right) \right)
$$

Now, changing the points $x_{k,r+2}^T$ for $x_{k,r+1}^T$ and redefining

$$
\tau_k := \max_{1 \leq i \leq r+2} \left\| x_{k,i}^T - x_k \right\|.
$$
we get

\[
\begin{pmatrix}
\nabla \phi_i \left( \frac{1}{k_{i,1}} \right)^T & - \nabla \phi_{r+1} \left( \frac{1}{k_{r,r+1}} \right)^T \\
\vdots & \vdots \\
\nabla \phi_r \left( \frac{1}{k_{r,r}} \right)^T & - \nabla \phi_{r+1} \left( \frac{1}{k_{r,r+1}} \right)^T \\
\end{pmatrix}
\begin{pmatrix}
\nabla \phi_i \left( \frac{1}{k_{i,1}} \right)^T \\
\vdots \\
\nabla \phi_r \left( \frac{1}{k_{r,r}} \right)^T \\
\end{pmatrix} H_k^{-1}
\begin{pmatrix}
\nabla \phi_i \left( \frac{1}{k_{i,1}} \right)^T & - \nabla \phi_{r+1} \left( \frac{1}{k_{r,r+1}} \right)^T \\
\vdots & \vdots \\
\nabla \phi_r \left( \frac{1}{k_{r,r}} \right)^T & - \nabla \phi_{r+1} \left( \frac{1}{k_{r,r+1}} \right)^T \\
\end{pmatrix}
\begin{pmatrix}
\lambda_i \\
\vdots \\
\lambda_r \\
\end{pmatrix}
\]

This last linear system yields

\[
\begin{pmatrix}
\nabla \phi_i \left( \frac{1}{k_{i,1}} \right)^T & - \nabla \phi_{r+1} \left( \frac{1}{k_{r,r+1}} \right)^T \\
\vdots & \vdots \\
\nabla \phi_r \left( \frac{1}{k_{r,r}} \right)^T & - \nabla \phi_{r+1} \left( \frac{1}{k_{r,r+1}} \right)^T \\
\end{pmatrix}
\begin{pmatrix}
\nabla \phi_i \left( \frac{1}{k_{i,1}} \right)^T \\
\vdots \\
\nabla \phi_r \left( \frac{1}{k_{r,r}} \right)^T \\
\end{pmatrix} H_k^{-1} \nabla \phi_{r+1} \left( \frac{1}{k_{r,r+1}} \right)^T + O \left( \frac{1}{k_{r,k}} \right).
\]

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

\[
\hat{\lambda}_{r+1} = 1 - \sum_{i=1}^{r} \hat{\lambda}_i.
\]

(37)
Therefore, considering $\lambda^* \in \mathbb{R}^{r+2}$ the solution of (36) and using equation (37), we must have

$$
\lambda^* = \begin{pmatrix}
\hat{\lambda}_1 \\
\vdots \\
\hat{\lambda}_r \\
\lambda^{r+1}_{r+1} \\
1 - \sum_{i=1}^{r} \lambda_i - \lambda^{r+1}_{r+1}
\end{pmatrix} + O\left(\tau_k, l_k\right)
$$

So, to complete our reasoning, we write the following relation between the primal-dual variables

$$
d_{k,l_k} = -H_k^{-1} \left[ \sum_{i=1}^{r+1} \lambda^*_i \nabla \phi_i \left( x_{k,i}^{l_k} \right) + \lambda^*_{r+2} \nabla \phi_{r+1} \left( x_{k,r+2}^{l_k} \right) \right] = -H_k^{-1} \left[ \sum_{i=1}^{r} \lambda^*_i \nabla \phi_i \left( x_{k,i}^{l_k} \right) + \left( \lambda^*_{r+1} + \lambda^*_{r+2} \right) \nabla \phi_{r+1} \left( x_{k,r+1}^{l_k} \right) \right] + O\left(\tau_k, l_k\right)
$$

Hence, $d_{k,l_k}$ is exactly the search direction obtained in (21) with an additional error vector. Therefore, the term $O\left(\tau_k, l_k\right)$ 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.

References

1. Balinski, M.L., Wolfe, P.: Nondifferentiable Optimization. Mathematical Programming Studies, vol. 3. North-Holland, Amsterdam (1975)
2. Bonnans, J.F., Gilbert, J.C., Lemaréchal, C., Sagastizábal, C.A.: Numerical Optimization: Theoretical and Practical Aspects, 2nd edn. Springer, Berlin (2006)
3. Boyd, S., V andenberghe, 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. Math. Oper. Res. 27(3), 567–584 (2002)
5. Burke, J.V., Lewis, A.S., Overton, M.L.: A robust gradient sampling algorithm for nonsmooth, non-convex optimization. SIAM J. Optim. 15(3), 751–779 (2005)
6. Clarke, F.H.: Optimization and Nonsmooth Analysis, vol. 5. SIAM, Montreal (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. Crema, A., Loreto, M., Raydan, M.: Spectral projected subgradient with a momentum term for the Lagrangean dual approach. Comput. Oper. Res. 34(10), 3174–3186 (2007)
9. Curtis, F.E., Overton, M.L.: A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization. SIAM J. Optim. 22(2), 474–500 (2012)
10. Curtis, F.E., Que, X.: An adaptive gradient sampling algorithm for non-smooth optimization. Optim. Methods Softw. 28(6), 1302–1324 (2013)
11. Curtis, F.E., Que, X.: A quasi-Newton algorithm for nonconvex, nonsmooth optimization with global convergence guarantees. Math. Program. Comput. 7(4), 399–428 (2015)
12. Daniilidis, A., Sagastizábal, C., Solodov, M.: Identifying structure of nonsmooth convex functions by the bundle technique. SIAM J. Optim. 20(2), 820–840 (2009)
13. Di Pillo, G., Grippo, L., Lucidi, S.: A smooth method for the finite minimax problem. Math. Program. 60(1), 187–214 (1993)
14. Do, T.M.T., Artières, T.: Regularized bundle methods for convex and non-convex risks. J. Mach. Learn. Res. 13(1), 3539–3583 (2012)
15. 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)
16. Du, D.Z., Pardalos, P.M.: Minimax and Applications, vol. 4. Springer, Boston (2013)
17. Fuduli, A., Gaudioso, M., Giallombardo, G.: A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. Optim. Methods Softw. 19(1), 89–102 (2004)
18. Gaudioso, M., Gorgone, E., Monaco, M.F.: Piecewise linear approximations in nonconvex nonsmooth optimization. Numerische Mathematik 113(1), 73–88 (2009)
19. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Rev. 47(1), 99–131 (2005)
20. Goldstein, A.A.: Optimization of Lipschitz continuous functions. Math. Program. 13(1), 14–22 (1977)
21. Griewank, A., Walther, A.: Evaluating Derivatives, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2008)
22. Grothey, A., McKinnon, K.: A superlinearly convergent trust region bundle method. Report, Department of Mathematics & Statistics, Edinburgh University (1998)
23. Haarala, M., Miettinen, K., Mäkelä, M.M.: New limited memory bundle method for large-scale non-smooth optimization. Optim. Methods Softw. 19(6), 673–692 (2004)
24. Helou, E.S., Santos, S.A., Simões, L.E.A.: On the differentiability check in gradient sampling methods. Optim. Methods Softw. 31(5), 983–1007 (2016)
25. Helou, E.S., Santos, S.A., Simões, L.E.A.: On the local convergence analysis of the gradient sampling method for finite max-functions. J. Optim. Theory Appl. 175(1), 137–157 (2017)
26. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms I. Springer, New York (1993)
27. Huber, G.: Gamma function derivation of n-sphere volumes. Am. Math. Mon. 89(5), 301–302 (1982)
28. Kelley Jr., J.E.: The cutting-plane method for solving convex programs. J. Soc. Ind. Appl. Math. 8(4), 703–712 (1960)
29. Kiwiel, K.C.: Methods of descent for nondifferentiable optimization, vol. 1133. Springer, Berlin (1995)
30. Kiwiel, K.C.: Restricted step and Levenberg–Marquardt techniques in proximal bundle methods for nonconvex nondifferentiable optimization. SIAM J. Optim. 6(1), 227–249 (1996)
31. Kiwiel, K.C.: Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization. SIAM J. Optim. 18(2), 379–388 (2007)
32. Lemaréchal, C., Mifflin, R.: Global and superlinear convergence of an algorithm for one-dimensional minimization of convex functions. Math. Program. 24(1), 241–256 (1982)
33. Lemaréchal, C., Oustry, F., Sagastizábal, C.: The U-Lagrangian of a convex function. Trans. Am. Math. Soc. 352(2), 711–729 (2000)
34. Lemaréchal, C., Sagastizábal, C.: Practical aspects of the moreau–yosida regularization: theoretical preliminaries. SIAM J. Optim. 7(2), 367–385 (1997)
35. Lewis, A.S.: Active sets, nonsmoothness, and sensitivity. SIAM J. Optim. 13(3), 702–725 (2002)
36. Lewis, A.S., Overton, M.L.: Nonsmooth optimization via quasi-Newton methods. Math. Program. 141(1–2), 135–163 (2013)
37. Loreto, M., Aponte, H., Cores, D., Raydan, M.: Nonsmooth spectral gradient methods for unconstrained optimization. EURO J. Comput. Optim. 5(4), 529–553 (2017)
38. Lukšan, L., Vlček, J.: A bundle-Newton method for nonsmooth unconstrained minimization. Math. Program. 83(1–3), 373–391 (1998)
39. Mäkelä, M.: Survey of bundle methods for nonsmooth optimization. Optim. Methods Softw. 17(1), 1–29 (2002)
40. Maratos, N.: Exact penalty function algorithms for finite dimensional and control optimization problems. Ph.D. thesis, Imperial College, London (1978)
41. Maréchal, P., Ye, J.J.: Optimizing condition numbers. SIAM J. Optim. 20(2), 935–947 (2009)
42. Mifflin, R., Sagastizábal, C.: VU-decomposition derivatives for convex max-functions. In: Théra, M., Tichatschke, R. (eds.) Ill-posed Variational Problems and Regularization Techniques. Lecture Notes in Economics and Mathematical Systems, vol. 477, pp. 167–186. Springer, Berlin (1999)
43. Mifflin, R., Sagastizábal, C.: A VU-algorithm for convex minimization. Math. Program. 104(2–3), 583–608 (2005)
44. Mifflin, R., Sagastizábal, C.: A science fiction story in nonsmooth optimization originating at IIASA. In: Grötschel, M. (ed.) Documenta Mathematica Optimization Stories, pp. 291–300. Deutschen Mathematiker-Vereinigung, Bielefeld (2012)
45. Miller, S.A., Malick, J.: Newton methods for nonsmooth convex minimization: connections among U-Lagrangian, Riemannian Newton and SQP methods. Math. Program. 104(2), 609–633 (2005)
46. Moreau, J.J., Panagiotopoulos, P.D.: Nonsmooth Mechanics and Applications, vol. 302. Springer, Vienna (2014)
47. Nocedal, J., Wright, S.: Numerical Optimization, 2nd edn. Springer, New York (2006)
48. Oliveira, W., Sagastizábal, C.: Bundle methods in the XXIst century: a bird’s-eye view. Pesquisa Operacional 34(3), 647–670 (2014)
49. 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)
50. Peng, C., Jin, X., Shi, M.: Epidemic threshold and immunization on generalized networks. Phys. A Stat. Mech. Appl. 389(3), 549–560 (2010)
51. Wang, F.C., Chen, H.T.: Design and implementation of fixed-order robust controllers for a proton exchange membrane fuel cell system. Int. J. Hydrog. Energy 34(6), 2705–2717 (2009)
52. Zhang, J., Kim, N.H., Lasdon, L.: An improved successive linear programming algorithm. Manag. Sci. 31(10), 1312–1331 (1985)