A NONMONOTONE APPROACH WITHOUT DIFFERENTIABILITY TEST FOR GRADIENT SAMPLING METHODS

ELIAS SALOMÃO HELOU*, SANDRA AUGUSTA SANTOS†, AND LUCAS E. A. SIMÕES‡

Abstract. Recently, optimization problems involving nonsmooth and locally Lipschitz functions have been subject of investigation, and an innovative method known as Gradient Sampling has gained attention. Although the method has shown good results for important real problems, some drawbacks still remain unexplored. This study suggests modifications to the gradient sampling class of methods in order to solve those issues. We present an alternative procedure that suppresses the differentiability test without affecting its convergence and we also exhibit a nonmonotone line search that can improve the robustness of these methods. Finally, we show some numerical results that support our approach.

Key words. nonsmooth optimization, nonconvex optimization, gradient sampling, nonmonotone line search, differentiability test

AMS subject classifications. 65K10, 90C26

1. Introduction.

The need to minimize a function $f : \mathbb{R}^n \to \mathbb{R}$ that is differentiable in a dense set of $\mathbb{R}^n$, but not in the full domain, arise in many areas of science [16, 18, 22]. Since a uniform random choice of points in $\mathbb{R}^n$ gives, with probability one, a point where f is differentiable, one could wrongly believe that standard algorithms for smooth minimization might work to solve this kind of problem. In fact, since many real problems have their minimum points where f is not differentiable, these algorithms have to face nondifferentiability, and in many cases fail to solve the optimization problem. Therefore, the study of specific methods is needed.

The development of the gradient sampling algorithm (GS) as a tool for solving unconstrained nonsmooth optimization problems was an important contribution to the field, primarily as a robust and practical algorithm, exhibiting meaningful numerical results for challenging problems [2, 3]. It can be seen as a generalization of the well established steepest descent method for smooth functions, especially after Kiwiel's introduction of the nonnormalized version of this method [12].

The main idea that allowed the emergence of convergence analysis for many methods that deal with locally Lipschitz functions was the notion of generalized gradients contained in Clarke's studies [4, 6]. Thenceforth, important methods were introduced in the literature. For example, the bundle and subgradient methods [1, 11, 20], and more recently, some variants of the GS method [7, 12], which were conceived to overcome both theoretical and practical limitations.

One drawback that the original authors pointed out [3] is that although the convergence analysis was obtained requiring a sufficient decrease of the objective function in each iteration, they observed that, in practice, the algorithm has a better performance if one asks just for $f(x_{k+1}) < f(x_k)$. The nonnormalized GS was a step further to surpass this problem, since near a stationary point it eases the line search. However, even in the nonnormalized version, it is possible to observe in real problems that

[*] Institute of Mathematical Sciences and Computation, University of São Paulo. This author was partially supported by FAPESP grants 2013/16508-3 and 2013/07375-0. (elias@icmc.usp.br).

[†] Department of Applied Mathematics, University of Campinas, Campinas, SP, Brazil. This author was partially supported by CNPq grant 304032/2010-7, FAPESP grants 2013/05475-7 and 2013/07375-0 and PRONEX Optimization. (sandra@ime.unicamp.br).

[‡] Department of Applied Mathematics, University of Campinas, Campinas, SP, Brazil. This author was supported by FAPESP grant 2013/14615-7. (simoes.lea@gmail.com).

the line search may fail during the execution of the algorithm, since the desired step size might be too small in order to satisfy the sufficient decrease condition

$$f(x_k + t_k d_k) \leq f(x_k) - t_k \rho(d_k), \tag{1.1}$$

where $\rho(d) : \mathbb{R}^n \to \mathbb{R}_+$ is a function defined accordingly for each GS variant. Consequently, it might result in a null step under finite precision arithmetic.

Another point that has not been solved is the need that each iterate x_k of gradient sampling methods based on standard backtracking line search must be in the differentiable set of the objective function. Although it is possible to guarantee this requirement when nondifferentiability is detected, for most real problems, it is impossible to test this condition, forcing the practical algorithm to disregard this verification. Therefore, a modification in the actual algorithm is needed to fill the gap between the theoretical and the practical algorithms.

It is important to mention that Kiwiel presented in [12] an algorithm that does not require the differentiability test. However, it is a method that does not guarantee that in each iteration the algorithm will produce a descent direction, since it does not use x_k to obtain the search direction. Hence, up to our knowledge, there is no method in literature that guarantees, with probability one, that the objective function will be differentiable in every x_k and does not produce null steps, unless one implements a differentiability test.

Our purpose in this paper is to solve the issue of the differentiability test and to change the standard line search found in gradient sampling methods by adding a positive scalar Δ_k on the right side of (1.1), and thus, relaxing the sufficient decrease condition. For the convergence analysis we require $\{\Delta_k\}$ to be a summable sequence.

This paper is organized as follows. In section 2, we present the basic concepts behind the GS methods and exhibit a model algorithm that encompasses most of these methods. Section 3 shows a strategy to avoid the differentiability test during the execution of the method by adding a perturbation vector in the search direction. In the same section, we propose a nonmonotone line search as a way to prevent tiny step sizes, and we present a convergence proof that embraces ours and several other well known GS methods. Section 4 contains the numerical results comparing the standard algorithms with our new strategies. Finally, Section 5 is left for the conclusions of this paper.

In order to facilitate the exposition of some results in this text, we use the following notations:

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

2. Basic Concepts.

This study focuses on the unconstrained optimization problem:

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz function and continuously differentiable in an open dense subset $\mathcal{D} \subset \mathbb{R}^n$. For all real-valued locally Lipschitz functions it is possible to define the Clarke's subdifferential set for f at x, which is a generalization of the derivative concept for smooth functions.

Definition 2.1 (Subdifferential set, subgradient) *The set given by*

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

is called the Clarke's subdifferential set for f at x. Moreover, any $v \in \overline{\partial} f(x)$ is known as a subgradient of f at x.

Although this is an important concept for the theory of nonsmooth functions, a more general set is useful, namely, the Goldstein's ϵ-subdifferential set.

Definition 2.2 (ϵ-Subdifferential set, ϵ-subgradient) *The ϵ-subdifferential set for f at x is given by*

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

Moreover, any $v \in \overline{\partial}_\epsilon f(x)$ is known as an ϵ-subgradient of f at x.

A close notion to the ϵ-subdifferential set is the multifunction defined as

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

for which it holds $\overline{\partial}_\delta f(x) \subset \mathcal{G}_\epsilon(x) \subset \overline{\partial}_\epsilon f(x)$, if $\epsilon > \delta \geq 0$. Unfortunately, none of these sets are easy to compute for general problems, and therefore, their use in algorithms is impracticable. For this reason, most of the methods that deals with nonsmooth functions requires only one element of these sets, or tries to approximate, in some sense, one of them. As shown in [2, Theorem 2.1], the gradient sampling method attempts to approximate the set $\mathcal{G}_\epsilon(x)$ using a sample of points in an ϵ-neighborhood of x and the same occurs for its variants. Moreover, if $x_k \in \mathcal{D}$, it is possible to obtain a descent direction with this approximated set.

Using the properties of projections, it is possible to see that if we project the null vector, with respect to some norm $\|\cdot\|$, onto the set of convex combinations of the gradient of the sampled points and the gradient of the current iteration $\nabla f(x_k)$, we obtain a descent direction for the function at x_k. Taking into account different norms for $\mathbb{R}^n$, one can produce different directions. Indeed, considering a positive definite symmetric matrix H_k for each iteration k of the method, and using $\|\cdot\|_{H_k}$ as the norm of the space, it is possible to recover several GS algorithms. For example, if we set $H_k = I$ for all k, we obtain the standard GS method. However, updating H_k, with limited memory LBFGS techniques, we get a variant of the methods suggested in [7].

Essentially, most of the gradient sampling methods that use backtracking line search to find a better point at each iteration follow the same steps, with some slight modifications for each algorithm. Broadly, we can define a model algorithm as presented in Algorithm 2.1.

One may be apprehensive with the requirement that the sample points must be in $\mathcal{D}$, but if we uniformly choose a point in the domain, with probability one this point will belong to $\mathcal{D}$. Therefore, with probability one, the aforementioned algorithm does not terminately prematurely. Moreover, for the convergence analysis, we assume from now on that:

```
Step 0. Set $k = 0$, $x_0 \in \mathcal{D}$, $m \in \mathbb{N}$ with $m \geq n + 1$, fixed real numbers $\nu_0, \epsilon_0 > 0$
        and $0 < \theta_\nu, \theta_\epsilon, \gamma, \beta < 1$.
Step 1. Choose $\{x_{k1}, \ldots, x_{km}\} \in \mathcal{B}(x_k, \epsilon_k)$ with randomly, independently and
        uniformly sampled elements. If $x_{ki} \in \mathcal{D}$, for all $i \in \{1, \ldots, m\}$,
        then go to Step 2. Otherwise, STOP!
Step 2. Set $G_k = [\nabla f(x_k)\ \nabla f(x_{k1})\ \ldots \nabla f(x_{km})]$ and find $g_k = H_k^{-1} u_k$
        such that $u_k = G_k \lambda_k$ and $\lambda_k$ solves

        (2.1)   $\min_{\lambda} \ \frac{1}{2} \lambda^T G_k^T H_k^{-1} G_k \lambda$
                $s.t. \ e^T \lambda = 1, \ \lambda \geq 0$

        where $H_k \in \mathbb{R}^{n \times n}$ is a predefined positive definite symmetric matrix.
Step 3. If $\min\{\|g_k\|, \|g_k\|_{H_k}\} < \nu_k$, then set $\epsilon_{k+1} = \theta_\epsilon \epsilon_k$, $\nu_{k+1} = \theta_\nu \nu_k$,
        $x_{k+1} = x_k$ 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

                $f(x_k + t_k d_k) \leq f(x_k) - \beta \alpha_k t_k g_k^T H_k g_k,$

        where $d_k = -\alpha_k g_k$, for some $\alpha_k > 0$.
Step 5. If $x_k + t_k d_k \in \mathcal{D}$, then set $x_{k+1} = x_k + t_k d_k$. Otherwise, find

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

        with $f(x_{k+1}) \leq f(x_k) - \beta \alpha_k t_k g_k^T H_k g_k$.
Step 6. Set $k \leftarrow k + 1$ and go back to Step 1.
```

Algorithm 2.1: Model algorithm for gradient sampling methods

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

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

In addition, the scalar $\alpha_k > 0$ must satisfy $\alpha_k \in \{1, \vartheta / \|g_k\|\}$, for all $k \in \mathbb{N}$, where ϑ is a fixed positive real number.

The first assumption on the matrix H_k is commonly used in quasi-Newton techniques and was recently used to generate a gradient sampling method with LBFGS approach [7]. The requirement that $\alpha_k \in \{1, \vartheta / \|g_k\|\}$ is just a form to embrace the nonnormalized and the normalized versions of GS, and therefore, the possibility that $\|g_k\| \to 0$ is not an issue.

3. New Strategies for GS Methods. In this section we present new strategies to overcome the well known gaps between the implemented and the theoretical algorithms, proposing a reliable convergent GS method.

3.1. Eliminating the Differentiability Test. Although Step 5 of the model algorithm is indispensable for the convergence proof of the existing gradient sampling methods, this step is not executed in practice, since a differentiability test is impractical in most of the real problems. This subsection has the intent to circumvent this matter by adding a perturbation vector in the search direction, ensuring, with probability one, that every x_k will be in the differentiable set. For this purpose, we suggest that Step 4 of Algorithm 2.1 is replaced by the following alternative procedure:

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

$$f(x_k + t_k d_k) \leq f(x_k) - \beta\alpha_k t_k g_k^T H_k g_k,$$

where $d_k = -\alpha_k(g_k + \xi_k)$ and

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

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

With this modification, we claim that Step 5 can be suppressed without affecting the convergence properties. To ensure this, we start proving a lemma.

Lemma 3.1 *If d_k is given by Step 4a and x_k is not a stationary point for f, then d_k is a descent direction for f at $x_k \in \mathcal{D}$ and with probability one $x_k + t_k d_k \in \mathcal{D}$, where t_k is the step size obtained with the Armijo line search.*

Proof. First, by relation (4.3) in [7, Lemma 4.3], we know that $\nabla f(x_k)^T g_k \geq \|g_k\|_{H_k}^2$. Therefore, it follows that

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

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

First, we define the following isomorphism:

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

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

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

Therefore, considering a uniform distribution, we see that

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

where we denote $\mathrm{Vol}(\mathcal{A})$ as the volume of $\mathcal{A} \subset \mathbb{R}^n$. Consequently, setting $\sigma = \alpha_k \gamma^j$ and $z = x_k - \sigma g_k$, we see that for a uniform random choice of $\xi_k \in \mathcal{B}(0,r)$, we have, for any $j \in \mathbb{N}$, that $\xi_k \in \hat{\mathcal{D}}$ with probability one, and then,

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

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

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

which completes the proof. □

According to this result, we can assure, with probability one, that the function f is differentiable for all x_k by adding a perturbation vector in the usual direction search. Hence, with probability one, the algorithm is still well defined if we suppress Step 5.

Now, we can proceed following closely to the results of [12]. We start with a slight modification of [12, Lemma 3.1]:

Lemma 3.2 *Suppose that $\mathcal{C} \subset \mathbb{R}^n$ is a nonempty compact set such that $0 \notin \mathcal{C}$. Thus, if $\beta \in (0,1)$ and $H \in \mathbb{R}^{n\times n}$ is a positive definite symmetric matrix, then there exists $\delta > 0$ such that $u, v \in \mathcal{C}$ and $\|u\|_H \le \mathrm{dist}_H(0,\mathcal{C}) + \delta$ imply $v^T H u > \beta\|u\|_H^2$. Moreover, for a fixed $u \in \mathcal{C}$ there exists $\mu > 0$ such that*

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

Proof. Following the same proof in [12, Lemma 3.1] but considering $\|\cdot\|_H$ and dist_H instead of $\|\cdot\|$ and dist_I, we have $v^T H u > \beta\|u\|_H^2$, for all $v \in \mathcal{C}$. Hence, since $\mathcal{C}$ is a compact set, we have that

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

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

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

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

and

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

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

Lemma 3.3 *Let $\epsilon > 0$, $\overline{x} \in \mathbb{R}^n$ and H be a positive definite symmetric matrix.*

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

ii) Assuming $0 \notin \mathcal{G}_\epsilon(\overline{x})$, pick $\delta > 0$ and $\beta \in (0,1)$ as in Lemma 3.2 for $\mathcal{C} := \mathcal{G}_\epsilon(\overline{x})$, $H = H_k^{-1}$ and then τ and $\overline{\mathcal{V}}$ as in statement (i). Suppose at iteration k of Algorithm 2.1, Step 5 is reached with $x_k \in \mathcal{B}(\overline{x}, \min\{\tau, \epsilon/3\})$, $\epsilon_k = \epsilon$ and $(x_{k1}, \ldots, x_{km}) \in \overline{\mathcal{V}}$. Then, $u_k \in \mathcal{C}$. Moreover, considering $u = u_k$ in Lemma 3.2 and selecting μ for this fixed u, we have that if $\|\xi_k\| < \mu/L$, where L is the Lipschitz constant over $\mathcal{B}(\overline{x}, \epsilon)$, then $t_k \geq \min\{1, \gamma\underline{\varsigma}\epsilon/(6L), \gamma\epsilon/(6\vartheta)\}$.

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

Proof. Assertions $i)$ follows immediately from [12, Lemma 3.2] assuming $\|\cdot\|_H$ and dist_H and $iii)$ has the same proof of [12, Lemma 3.2] as well. Then, let us prove $ii)$, since it has some minor modifications.

By hypothesis, we have that $(x_{k1}, \ldots, x_{km}) \in \overline{\mathcal{V}} \subset \mathcal{V}_\epsilon^{H_k^{-1}}(\overline{x}, x, \delta)$. Therefore, it follows that $\operatorname{dist}_{H_k^{-1}}(0, \operatorname{co}\{\nabla f(x_{ki})\}_{i=1}^m) \leq \rho_\epsilon^{H_k^{-1}}(\overline{x}) + \delta$ and $\operatorname{co}\{\nabla f(x_{ki})\}_{i=1}^m \subset \mathcal{G}_\epsilon(\overline{x})$. Now, by the manner u_k is computed in Step 2 and as $\nabla f(x_k) \in \mathcal{G}_\epsilon(\overline{x})$ (since $x_k \in \mathcal{B}(\overline{x}, \min\{\tau, \epsilon/3\}) \cap D)$, we have that $u_k \in \mathcal{G}_\epsilon(\overline{x})$ (which also gives us that $\|u_k\| \leq L$) and $\|u_k\|_{H_k^{-1}} \leq \rho_\epsilon^{H_k^{-1}}(\overline{x}) + \delta$. Hence, by Lemma 3.2, there exists $\mu > 0$ such that

$$\mu = \inf_{v \in \mathcal{G}_\epsilon(\overline{x})} v^T H_k^{-1} u_k - \beta \|u_k\|_{H_k^{-1}}. \tag{3.1}$$

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

$$-\beta\gamma^{-1}\alpha_k t_k g_k^T H_k g_k \leq f(x_k + \gamma^{-1}t_k d_k) - f(x_k). \tag{3.2}$$

But we know, from the generalized mean value theorem for Lipschitz functions [5, Theorem 2.3.7], that there exist $y^k \in [x_k + \gamma^{-1}t_k d_k, x_k]$ and $v_k \in \overline{\partial} f(y_k)$ such that

$$f(x_k + \gamma^{-1}t_k d_k) - f(x_k) = \gamma^{-1}t_k v_k^T d_k. \tag{3.3}$$

On the other hand, we observe that

$$\begin{aligned} \gamma^{-1}t_k\|d_k\| &\leq \gamma^{-1}t_k\alpha_k(\|g_k\| + \|\xi_k\|) \\ &\leq 2\gamma^{-1}t_k\alpha_k\|g_k\|. \end{aligned}$$

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

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

otherwise, if $\alpha_k\|g_k\| = \vartheta$, then $\gamma^{-1}t_k\|d_k\| \leq \gamma^{-1}t_k 2\vartheta < \epsilon/3$. Thus, since $\|x_k - \overline{x}\| \leq \epsilon/3$, we have that $v_k \in \mathcal{G}_\epsilon(\overline{x})$ and also $\|v_k\| \leq L$. Now, by (3.2) and (3.3), we have

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

which is a contradiction with (3.1). Therefore, we have the desired lower bound for t_k. □
With this result in hands we are ready to prove the convergence of the model algorithm presented in the previous section.

Theorem 3.1 *If $\{x_k\}$ is a sequence generated by Algorithm 2.1 with Step 4a, then either $f(x_k) \to -\infty$ or every cluster point of $\{x_k\}$ is a stationary point for f.*

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

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

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

$$\sum_{k=1}^{\infty} \alpha_k t_k g_k^T H_k g_k < \infty. \tag{3.4}$$

We also have by Assumption 2.1 that if $\|g_k\| \neq 0$, then

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

and therefore, this inequality together with (3.4), give us

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\|\|g_k\| < \infty. \tag{3.5}$$

Now, we break the proof in two cases:

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

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

In the first case, we have that $\|g_k\|, g_k^T H_k g_k \geq \nu$, for all k sufficiently large. By (3.5), it implies that the whole sequence must converge, that is, $x_k \to \overline{x}$, for some $\overline{x} \in \mathbb{R}^n$. Considering that $L > 0$ is the Lipschitz constant of f over the set $\mathcal{B}(\overline{x}, \epsilon)$, we see that $\|g_k\| \leq L$ for all $k \geq k_1$, with a sufficient large $k_1 \in \mathbb{N}$. Hence, since $\alpha_k \in \{1, \vartheta/\|g_k\|\}$, we have the lower bound $\alpha_k \geq \min\{1, \vartheta/L\}$, for all $k \geq k_1$. Using this information together with (3.4) and $g_k^T H_k g_k \geq \nu$, we have that $t_k \to 0$.

If $0 \notin \mathcal{G}_\epsilon(\overline{x})$ there exist δ, τ, μ and $\overline{\mathcal{V}}$ as in Lemma 3.3. Moreover, since ξ_k is uniformly sampled, there exists, with probability one, an infinite set $\mathcal{K} \subset \mathbb{N}$ such that $\|\xi_k\| < \mu/L$, for all $k \in \mathcal{K}$. Now, since $t_k \to 0$ and $x_k \to \overline{x}$, there exists k_2 such that $x_k \in \mathcal{B}(\overline{x}, \min\{\tau, \epsilon/3\})$ and $t_k < \min\{1, \gamma\underline{\varsigma}\epsilon/(6L), \gamma\epsilon/(6\vartheta)\}$, for all $k \in \mathcal{K}$ and $k \geq k_2$. This implies that $(x_{k1}, \ldots, x_{km}) \notin \overline{\mathcal{V}}$ for all $k \in \mathcal{K}$ and $k \geq k_2$, which is an event that has probability zero to occur.

On the other hand, if $0 \in \mathcal{G}_\epsilon(\overline{x})$, we can choose $\delta = \nu/2$, for Lemma 3.3 i), and pick $k_3 \in \mathbb{N}$ such that $k \geq k_3$ implies that $x_k \in \mathcal{B}(\overline{x}, \tau)$. So, we have that

$$\nu \leq \|g_k\|_{H_k} \leq \operatorname{dist}_{H_k}\left(0, \operatorname{co}\{\nabla f(x_{ki})\}_{i=1}^m\right), \text{ for all } k \geq k_3,$$

and consequently, $(x_{k1}, \ldots, x_{km}) \notin \overline{\mathcal{V}}$ for all $k \geq k_3$. Again, this is an event that has probability zero to happen. Therefore, with probability one, we must have $\epsilon_k \to 0$.

In the last case, we can follow the same steps in the final part (referred as iii) of the proof in [12, Theorem 3.3] to show that, with probability one, every cluster point is a stationary point for the function f. □

With the above result we complete the convergence analysis of the model algorithm modified with Step 4a and guarantee a practical procedure to ensure, with probability one, that f will be differentiable at any x_k. Furthermore, we presented a general convergence proof that embraces several gradient sampling methods, including the algorithms proposed in [3, 7] (with or without the normalization of the search direction).

Observation on the adaptive case. It is important to note that an adaptive approach, in the way Curtis and Que did in [7], can also be introduced in our algorithm without affecting the proofs presented in this section, by just noting that if we infinitely do incomplete line searches during the execution of the algorithm, then the case *i*) of the proof of Theorem 3.1 can not occur. Indeed, if case *i*) happens, then we must have that $t_k \to 0$, and since an incomplete line search presents a lower bound for the step size, it is impossible to have $t_k \to 0$. So, we rely in case *ii*) and with the same proof we see that $\overline{x}$ is a stationary point for f. Otherwise, if at some point of the algorithm, we no longer do incomplete line searches, then from a sufficiently large $k \in \mathbb{N}$ onwards, the algorithm behaves exactly like Algorithm 2.1, and thus, the same proof holds.

3.2. Nonmonotone Line Search. As was shown in Lemma 3.3, a lower bound for the step size t_k exists when the sample points of the current iteration lie in a specific open set. However, we can only assure that it will happen eventually, and therefore, the algorithm could perform many iterations without reaching this specific set. Consequently, due to computer rounding errors we may fail to find a step size t_k greater than zero, which is an undesirable behavior. For this reason, we propose, in this subsection, a nonmonotone line search for GS methods in order to overcome this drawback.

It is important to notice that in the smooth optimization field, the nonmonotone line search is used with a different purpose. Indeed, in such a context, this artifice is used to give more efficiency to the algorithm and avoid some "traps" of the function, being desirable to reduce the nonmonotone term when we are close to a stationary point. However, our aim is to have a more robust algorithm, and to that goal, we must use the nonmonotone tool when we are approaching a stationary point, since that is where the specific open set might be hard to reach. Once enlightened our purpose, we are ready to present an alternative to Step 4a, which not only eliminates the differentiability test, but also relaxes the requirement of the standard line search:

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

$$f(x_k + t_k d_k) \leq f(x_k) - \beta \alpha_k t_k g_k^T H_k g_k + \Delta_k,$$

where $\{\Delta_k\}$ is a summable positive sequence and $d_k = -\alpha_k(g_k + \xi_k)$, with

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

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

Since we request that $\sum \Delta_k < \infty$, the convergence proof for the Algorithm 2.1 with Step 4b is essentially the same found in Theorem 3.1:

Theorem 3.2 *If $\{x_k\}$ is a sequence generated by Algorithm 2.1 with Step 4b, then either $f(x_k) \to -\infty$ or every cluster point of $\{x_k\}$ is a stationary point for f.*

Proof. First, we observe that the lower bound for t_k found in Lemma 3.3 *ii*) is still valid, since every step size that satisfies the standard line search also satisfies the nonmonotone line search. Therefore, we can follow exactly the same proof in Theorem 3.1 by just noting that the inequalities (3.4) and (3.5) still hold, since $\{\Delta_k\}$ is a summable sequence. □

We have shown that with our new strategies to avoid the differentiability test and tiny step sizes during the algorithm we still have the convergence of important GS methods, and moreover, we presented a more reliable algorithm. For a practical validation of these amendments, in the next section we exhibit some numerical results for several problems from the literature.

4. Numerical Results. In order to observe the numerical behavior when one introduces a nonmonotone line search in algorithms that use gradient sampling as a main tool, we have solved three different classes of problems. The first class has 26 test problems, the same found in [7]. The second and the third classes[1] can be seen in [3] and are known to be challenging problems. For all tests, we have used the same instructions in the mentioned source papers, except for some slight modifications that are stressed out throughout this section.

The methods based on gradient sampling employed to obtain the numerical results are: (*i*) the original method (GS) proposed in [3], which uses a normalized

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

search direction; (*ii*) a not normalized version suggested by Kiwiel (K-GS) in [12], and (*iii*) one adaptive variant (AGS-LBFGS) presented in [7], which uses LBFGS updates to find a descent direction for the algorithm. All the tests were performed using Matlab (version R2012a) in an Intel Core 2 Duo T6500, 2.10 GHz and 4 Gb of RAM. We have used `quadprog` as the tool to solve the quadratic minimizations needed in each iteration, setting `interior-point-convex` as the algorithm choice and 10^{-12} as the tolerances TolX and TolFun and 10^{-8} (default value) as TolCon. The same quadratic solver was used for all methods. Since our aim for these numerical tests is to recognize a possible advantage to use nonmonotone line search and the quadratic solver has no influence in this goal, we did not implement the solver suggested in [7].

For the generation of the sequence $\{\Delta_k\}$, we have used the Zhang and Hager's nonmonotone line search [23]. Therefore, we set $\Delta_k = C_k - f(x_k)$, where we define

$$\eta_k \in [\eta_{\min}, \eta_{\max}] \text{ with } 0 \leq \eta_{\min} \leq \eta_{\max} < 1;$$

$$Q_0 = 1 \text{ and } Q_{k+1} = \eta_k Q_k + 1;$$

$$C_0 = f(x_0) \text{ and } C_{k+1} = (\eta_k Q_k C_k + f(x_{k+1}))/Q_{k+1}.$$

Under the hypothesis that $\eta_{\max} < 1$, it is possible to prove that $\{\Delta_k\}$ is a summable and positive sequence (see [19]). Finally, to indicate our versions of the gradient sampling methods (changing Step 4 by the alternative procedure Step 4b and suppressing Step 5), for each GS variant we added a prefix NM in each method name.

4.1. Class of mixed problems. This class of tests has 26 different unconstrained nonsmooth minimization problems, some of them being also nonconvex. These functions were suggested in [7] and they are a collection of problems introduced in [9, 10, 14, 15, 21]. The algorithms K-GS and AGS-LBFGS were used to solve these test problems as well as its nonmonotone versions NM-K-GS and NM-AGS-LBFGS.

We start our algorithms with $\epsilon_0 = 10^{-1}$ and $\nu_0 = \sqrt{10} \cdot 10^{-1}$ and we established, respectively, a lower bound of 10^{-12} and $\sqrt{10} \cdot 10^{-12}$ for ϵ_k and ν_k, for all k. Moreover, we have used $m = 2n$, $\theta_\nu = \theta_\epsilon = 10^{-1}$, $\gamma = 0.5$ and $\beta = 10^{-8}$ in both methods. It is important to say that we have followed the same parameter settings of [7] for the algorithmic choices and the input parameters of the test functions.

There are only two modifications that we need to point out. The first one is about the initial point x_0. Each initial point was randomly chosen to be a point in the n-dimensional Euclidean ball $\mathcal{B}(x_0', 1)$, where the points x_0' were the same picked in [7], with the exception that, for each solved optimization problem, we guaranteed that $x_0 = x_0'$ at least once. The second modification that we have made is on the actualization of the matrix H_k in the AGS-LBFGS method. In [7, Lemma 4.7], the authors proved a result similar to Lemma 3.3 *i*) presented in this paper. They proved (as we also did) that for a fixed H_k there will exist an open set $\overline{\mathcal{V}}$ such that if we sample the points in this set some desirable properties are obtained. However, for their updating of H_k, this matrix is dependent on the sample points of the current iteration, which is a circular reasoning. A way to avoid this issue and guarantee the convergence of the method is to have points for generating H_k that are independent from those that produce the matrix G_k. Therefore, we updated the matrix H_k using the same rules as in [7], but using the sample points obtained in the previous iteration, with the exception of the first iteration, where we have used $H_0 = I$. This corroborates the theoretical results, since with this new update of H_k we were able to obtain better

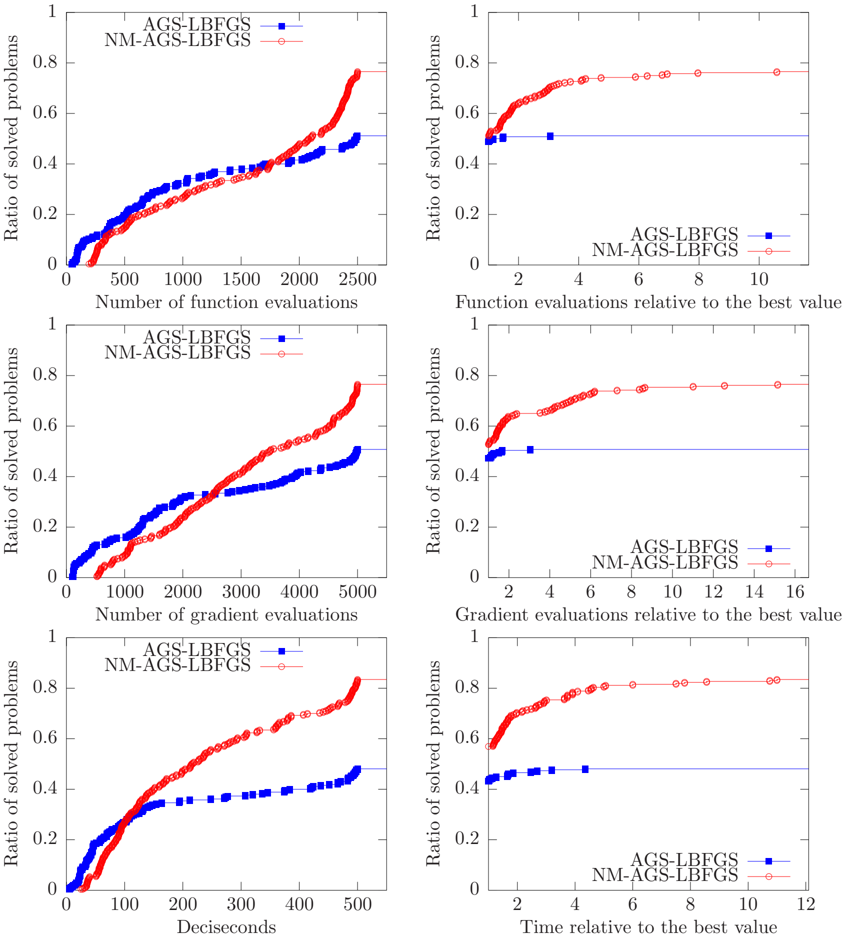

Fig. 1: On the left column we have data profiles comparing AGS-LBFGS with its nonmonotone version and on the right column we present the corresponding performance profiles. On the first row we have established a limit over the functions evaluations whereas on the second and third row we have, respectively, the number of gradient evaluations and CPU time as the budgets.

performance of the AGS-LBFGS method than with the originally proposed update. Nevertheless, it is worth mentioning that even using the update from [7], the profiles obtained remain similar to those exposed in this subsection.

The parameter values used for our nonmonotone approach were $\eta_k = 0.85$, $\forall k$, for the nonmonotone version of AGS-LBFGS and

$$\eta_0 = 0 \quad \text{and} \quad \eta_k = 0.85 \cdot \min\{-\log_2(t_{k-1})/25, 1\}, \quad \text{for } k = 1, 2, \dots, \tag{4.1}$$

for the nonmonotone version of K-GS. Moreover, we have used $c = 10^{-6}$ to produce the perturbation vector ξ_k in Step 4b. Finally, for AGS-LBFGS and NM-AGS-LBFGS methods we set the adaptive parameters as in [7].

As the tools for the comparative analysis, we have used performance profiles [8] and data profiles [17] with convergence tolerance of $\tau = 10^{-4}$ to observe how a non-

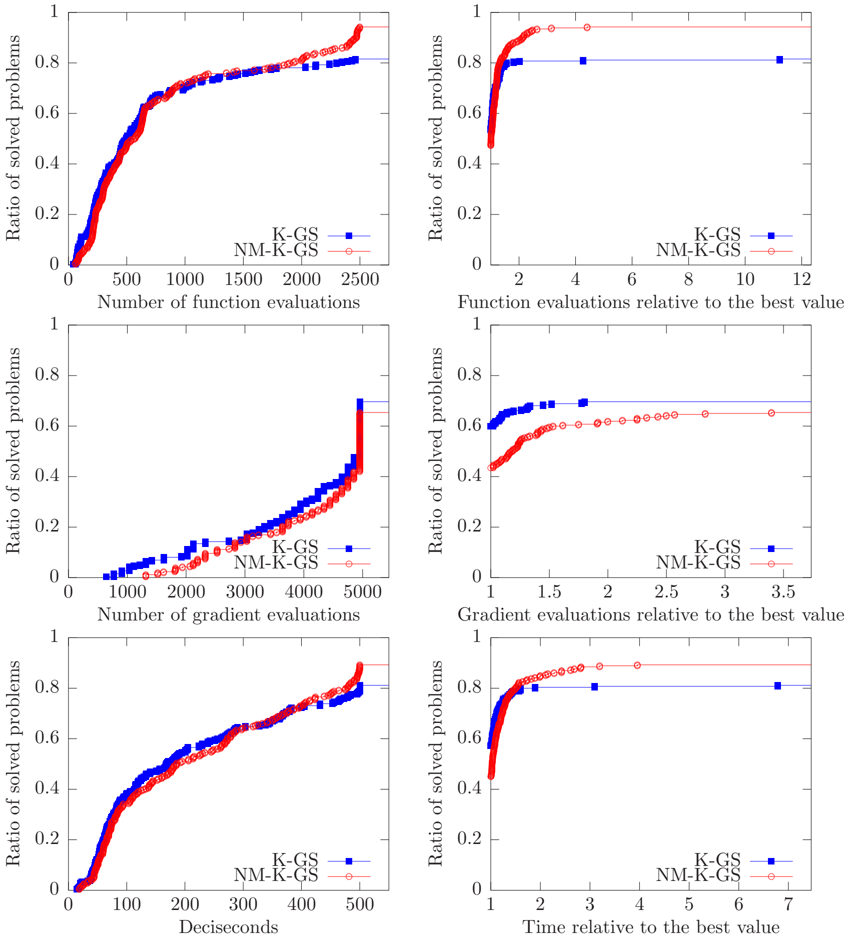

Fig. 2: On the left column we have data profiles comparing K-GS with its nonmonotone version and on the right column we present the corresponding performance profiles. On the first row we have established a limit over the functions evaluations whereas on the second and third row we have, respectively, the number of gradient evaluations and CPU time as the budgets.

monotone line search could improve the performance of the existing methods. We take our attention to three separate budgets: in the first, we limited our function evaluations to be $50n$, in the second one, we established $100n$ gradient evaluations and lastly, we have used n seconds as the maximum CPU time (which gives at least $4000 \cdot n^3$ basic arithmetic operations), where n is the dimension of the corresponding test problem. For all the budgets we looked to the function value to distinguish when a method was superior to the other one. Figure 1 shows the performance profiles and data profiles for the AGS-LBFGS method with its respective nonmonotone version and the same occurs for Figure 2, but for the K-GS method and its nonmonotone counterpart.

In both profiles it is possible to see an advantage in the use of nonmonotone line search. In practice, since this line search is more tolerant to accept a step size than the standard one, the nonmonotone version was able to reach a better function value with the same budget. Moreover, this new line search improves the main goal of adaptive methods, since it allows more incomplete Armijo's line search to succeed, and consequently, the algorithm tends to use fewer gradient evaluations in each iteration.

4.2. Product of eigenvalues. Now, we focus our attention on a specific problem that appears in the class of tests presented in the previous subsection: the minimization of eigenvalue products. All the instances of this problem were proposed in [3].

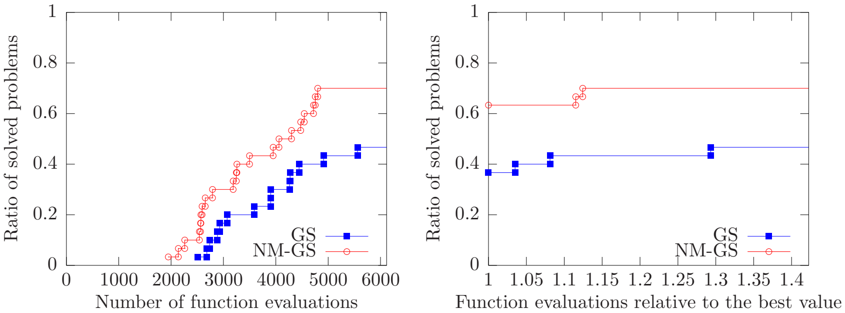

Fig. 3: On the left we have a data profile comparing GS with its nonmonotone version and on the right we present the corresponding performance profile. The budget for both profiles was the number of function evaluations and they present the results just for the instances $(N = 12,\, k = 6)$, $(N = 14,\, k = 7)$ and $(N = 16,\, k = 8)$.

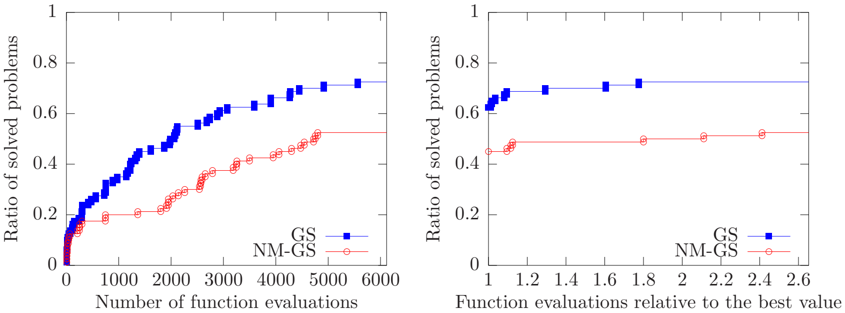

Fig. 4: On the left we have a data profile comparing GS with its nonmonotone version and on the right we present the corresponding performance profile. The budget for both profiles was the number of function evaluations and they present the results for all the instances.

Given a positive semidefinite symmetric matrix $A \in \mathbb{R}^{N\times N}$, we aim to find a positive semidefinite symmetric matrix $X \in \mathbb{R}^{N\times N}$ with ones on its diagonal such that the product of the largest k eigenvalues of the matrix $A \circ X$ (the componentwise product) is minimized in a problem with dimension $n = N(N-1)/2$. We have solved eight different instances of this optimization problem: $(N = 2, k = 1)$, $(N = 4, k = 2)$,...,$(N = 16, k = 8)$. We have solved each instance 10 times.

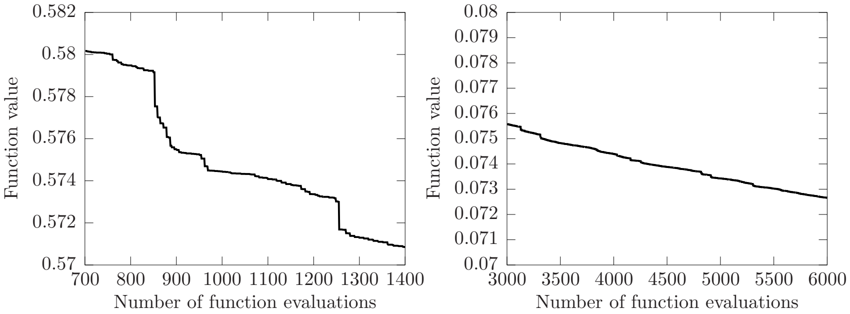

Fig. 5: The left figure presents, for the instance ($N = 8, k = 4$) and the GS method, the function value evolution in the last 50% of the budget of function evaluations whereas the right figure presents the same results for ($N = 16, k = 8$).

In practice, one can minimize the following nondifferentiable function:

$$f(x) = \prod_{i=1}^{k} \lambda_i(A \circ X) - \rho \cdot p(x),$$

where $\lambda_i(M)$ means the i-th eigenvalue (with a decreasing ordering) of the matrix M and $p(x) = \min\{0, \lambda_N(X)\}$. The function p plays the role of a penalization in case X is not positive semidefinite. Consequently, ρ is the penalization parameter, which we set $\rho = 100$ during all the instances.

Since the GS method was successful to reach the solution of all instances in [3], we analyzed the benefits of the nonmonotone line search on the number of function evaluations. For this purpose, all profiles presented in this subsection were generated comparing the function values over a budget of $50n$ function evaluations and have $\tau = 10^{-6}$ as the convergence tolerance, to capture better the attained precision of this particular problem. Furthermore, we set $\epsilon_0 = 10^{-1}$ and $\nu_0 = \sqrt{10} \cdot 10^{-1}$, establishing a lower bound of 10^{-6} and $\sqrt{10} \cdot 10^{-6}$ for ϵ_k and ν_k, respectively. For the other parameters, we set $m = 2n$, $\theta_\nu = \theta_\epsilon = 10^{-1}$ and $\gamma = 0.5$. Lastly, we have used $\beta = 0$ for the original GS method (since this value was recommended by [3]) and $\beta = 10^{-8}$ for the nonmonotone version NM-GS.

It is important to mention that unlike the original authors did in [3], we have not used a limit number on the iterations per radius. Since our goal is to generate profiles of both methods, it seemed fair to abdicate of this safeguard. Finally, we have used the same dynamical η_k defined in (4.1) for the nonmonotone method.

Figure 3 presents two data profiles of the last three instances, that is, the problems with biggest dimensions. It is possible to see that the addition of the nonmonotone term enhances the performance of GS, allowing it to reach a better function value with the same budget. However, Figure 4 shows an opposite result, with data and performance profiles that embrace all the instances. This behavior can be explained by the illustrative example of Figure 5. We notice that, for $N = 8$, the original method does not have difficulty to obtain better solutions along the last function evaluations. However, for $N = 16$, we can see that GS presents much more difficulty to reduce the function value. It is for situations like the one depicted at the right plot of Figure 5 that we expect the nonmonotone line search to be effective. When the complete set of instances is jointly analyzed (Figure 4), the behavior illustrated at the left plot of

Figure 5 dominates, and the nonmonotone strategy does not play a significant role. In fact, one can tune the parameter η_k such that the NM-GS method will present a better performance for the first instances, but the contribution of the nonmonotone term has to be small, that is, η_k will be closer to zero than to one.

4.3. Stability problem of a Boeing 767. This problem was presented and for the first time solved in [3]. It is a real problem that comes from the design optimization of a controller of an airplane (Boeing 767) at flutter condition. To a better understanding of the optimization problem involved in this situation, we must describe some stability measures of a dynamical system $\dot{x} = Ux$, where U is a squared matrix. However, we refer the reader to look at [3] to have a more complete introduction on the matter.

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

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

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

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

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

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

where $A \in \mathbb{R}^{55\times 55}$, $B \in \mathbb{R}^{55\times 2}$ and $C \in \mathbb{R}^{2\times 55}$ are fixed matrices and $X_1 \in \mathbb{R}^{2\times 2}$, $X_2 \in \mathbb{R}^{2\times k}$, $X_3 \in \mathbb{R}^{k\times 2}$ and $X_4 \in \mathbb{R}^{k\times k}$ are variable matrices.

For this problem we have used the algorithms GS and NM-GS (with η_k as in (4.1)) as solvers. We set $m = 2n$, $\theta_\nu = 1$, $\theta_\epsilon = 10^{-1}$, $\nu_0 = 10^{-6}$, $\epsilon_0 = 10^{-1}$ and $\gamma = 0.5$. Moreover, we have used perturbations proportional to the solution obtained by the NM-GS method for the spectral abscissa minimization problem in order to produce the initial starting points x_0. Since the original authors of [3] obtained better results with $\beta = 0$, we set the same value for GS method. On the other hand, for NM-GS we have used $\beta = 10^{-8}$. Furthermore, if $\epsilon_k < 10^{-6}$, we declared that the problem was solved, stopping the algorithm. We also kept safeguards, limiting the number of iterations per radius by 1000 (therefore the maximum number of iterations was 6000) and, if the line search fails, we skip the current radius and reduce it to the next radius. We justify these choices as follows: unlike the previous problem, our aim here was to look the final function value instead of profiles, since, in this particular problem, robustness is far more relevant than any other measure. Consequently, the introduction of this safeguard is not unfair to any method, as we are only concerned with the final function value.

Finally, we want to observe that this particular problem does not have a locally Lipschitz function. Therefore, our alternative procedure that adds the perturbation vector to the search direction to suppress the differentiability test can no longer be applied here, since its convergence proof demands a Lipschitz constant. Thus, we did not perturb the search direction during these tests.

GS				NM-GS			
k	f	$\|g\|$	it	k	f	$\|g\|$	it
0	7.86418e-05	2.2e-02	284	0	7.91388e-05	9.7e-07	187
1	1.04304e-04	4.2e-06	458	1	1.04520e-04	2.8e-06	699
2	1.05126e-04	5.1e-06	1235	2	1.06428e-04	4.2e-06	2411

Table 1: On the left we have the results of GS method and on the right the results of NM-GS for three instances of the problem. The columns labeled by f contain the best function value reached. The ones labeled by $\|g\|$ show the optimality residual norm of the last iteration and columns labeled by "it" exhibit the number of iterations needed to run the tests.

Table 1 reveals the results obtained by both methods for three instances of the problem, with dimensions respectively given by 4, 9 and 16. We have solved each instances 10 times, and we exhibit the best result obtained. It is possible to see that in all the instances the nonmonotone approach was more robust than the original method, relaxing the line search and allowing bigger step sizes. In addition, it is worth mentioning that not only the best result of each instance was better for NM-GS, but also in approximately 86% of the runs, the NM-GS algorithm was able to reach a better function value compared to the GS method. The reason for this positive results is that the nonmonotone approach enabled the algorithm to work with few line search failures, which allowed the method to reach a better solution and even a certificate of optimality (for the instance $k = 0$). Lastly, the more important result is that we were able to obtain better function values even with $\beta > 0$. This was prohibitive in the original method, since it would led to many line search failures, preventing the achievement of significant results.

5. Conclusion. In this paper we have presented a model algorithm for the well known class of gradient sampling methods and pursued ways to overcome important drawbacks of these algorithms. The differentiability test in Step 5 has always been a theoretical trick to guarantee the convergence of the methods, but none of the practical algorithms currently existing had this step implemented in their routines, as such kind of verification is impossible for general problems. We have presented a way to avoid this issue by adding a perturbation vector in the search direction of each iteration. It was shown that, with probability one, all the iterates x_k remain in the differentiable set, and consequently, such a test can be suppressed without affecting the convergence proof.

We also highlighted the perennial issue of tiny steps during the execution of these methods, specially in solving some challenging problems. This difficulty shows up due to the nature of these methods, which are random by design. To guarantee a lower bound for the step size t_k, we must sample points in a certain open set. Unfortunately, we can only assure that this event will happen, but we cannot know how many iterations this could take. As a result, the algorithms may need tiny step sizes to satisfy the backtracking inequality, and due to rounding errors, this eventually generates null steps. Consequently, the algorithms make efforts to produce tiny (or even none) improvements. For this issue, we have introduced a nonmonotone line search, which relaxes the backtracking inequality requirement. Moreover, this tool also supports the assessment of the differentiability, because even in the unlikely event that $x_k \notin \mathcal{D}$, the positive term Δ_k in the backtracking inequality will allow a successful line search.

Finally, we presented some numerical results that corroborate our theoretical amendments, showing that the nonmonotone line search in fact produces a more robust algorithm. A remarkable result was the improvement that we had in the challenging problem of a Boeing 767 at flutter condition, which does not have a locally Lipschitz objective function. It was shown that with the nonmonotone line search the extra freedom allows the algorithm to keep working and find a better solution than the one obtained with our implementation of the gradient sampling method.

We also would like to notice that we have not explored the possibility of using the nonmonotone approach in the method developed by Kiwiel in [13], since it would be out of the scope of this study. However, we see no reason to believe that the nonmonotone line search would fail to increase the performance and robustness of this technique too.

Acknowledgements. The authors would like to thank Frank E. Curtis and Xiaocun Que for kindly provide the implemented functions that compose the set of 26 problems addressed in this paper.

REFERENCES

[1] Michel L. Balinski and Philip Wolfe, *Nondifferentiable Optimization*, vol. 3, Mathematical Programming Studies., 1975.

[2] James V. Burke, Adrian S. Lewis, and Michael L. Overton, *Approximating subdifferentials by random sampling of gradients*, Mathematics of Operations Research, 27 (2002), pp. 567–584.

[3] ———, *A robust gradient sampling algorithm for nonsmooth, nonconvex optimization*, SIAM Journal on Optimization, 15 (2005), pp. 751–779.

[4] Frank H. Clarke, *Generalized gradients and applications*, Transactions of the American Mathematical Society, 205 (1975), pp. 247–262.

[5] ———, *Optimization and nonsmooth analysis*, vol. 5, SIAM, 1990.

[6] ———, *Nonsmooth analysis and control theory*, vol. 178, Springer, 1998.

[7] Frank E. Curtis and Xiaocun Que, *An adaptive gradient sampling algorithm for non-smooth optimization*, Optimization Methods and Software, 28 (2013), pp. 1302–1324.

[8] Elizabeth D. Dolan and Jorge J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91 (2002), pp. 201–213.

[9] Chen Greif and James M. Varah, *Minimizing the condition number for small rank modifications*, SIAM Journal on Matrix Analysis and Applications, 29 (2006), pp. 82–97.

[10] Marjo Haarala, Kaisa Miettinen, and Marko M. Mäkelä, *New limited memory bundle method for large-scale nonsmooth optimization*, Optimization Methods and Software, 19 (2004), pp. 673–692.

[11] Krzysztof C. Kiwiel, *Methods of descent for nondifferentiable optimization*, vol. 1133, Springer-Verlag, 1985.

[12] ———, *Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization*, SIAM Journal on Optimization, 18 (2007), pp. 379–388.

[13] ———, *A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization*, SIAM Journal on Optimization, 20 (2010), pp. 1983–1994.

[14] Adrian S. Lewis and Michael L. Overton, *Nonsmooth optimization via quasi-Newton methods*, Mathematical Programming, 141 (2013), pp. 135–163.

[15] L. Lukšan, M. Tuma, J. Vlcek, N. Ramešová, M. Šiška, J. Hartman, and C. Matonoha, *Ufo 2011–interactive system for universal functional optimization*, tech. report, 2011.

[16] Pierre Maréchal and Jane J. Ye, *Optimizing condition numbers*, SIAM Journal on Optimization, 20 (2009), pp. 935–947.

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

[18] Chengbin Peng, Xiaogang Jin, and Meixia Shi, *Epidemic threshold and immunization on generalized networks*, Physica A: Statistical Mechanics and its Applications, 389 (2010), pp. 549–560.

[19] Ekkehard W. Sachs and Stephen M. Sachs, *Nonmonotone line searches for optimization algorithms.*, Control & Cybernetics, 40 (2011), pp. 1059–1075.

[20] Naum Z. Shor, Krzysztof C. Kiwiel, and Andrzej Ruszcayski, *Minimization methods for non-differentiable functions*, Springer-Verlag New York, Inc., 1985.

[21] Anders Skajaa, *Limited memory BFGS for nonsmooth optimization*, Master's thesis, Courant Institute of Mathematical Science, New York University, (2010).

[22] Fu-Cheng Wang and Hsuan-Tsung Chen, *Design and implementation of fixed-order robust controllers for a proton exchange membrane fuel cell system*, International Journal of Hydrogen Energy, 34 (2009), pp. 2705–2717.

[23] Hongchao Zhang and William W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM Journal on Optimization, 14 (2004), pp. 1043–1056.