Smoothing and Worst-Case Complexity for Direct-Search Methods in Nonsmooth Optimization

R. Garmanjani * L. N. Vicente†

September 13, 2012

Abstract

In the context of the derivative-free optimization of a smooth objective function, it has been shown that the worst case complexity of direct-search methods is of the same order as the one of steepest descent for derivative-based optimization, more precisely that the number of iterations needed to reduce the norm of the gradient of the objective function below a certain threshold is proportional to the inverse of the threshold squared.

Motivated by the lack of such a result in the non-smooth case, we propose, analyze, and test a class of smoothing direct-search methods for the unconstrained optimization of non-smooth functions. Given a parameterized family of smoothing functions for the non-smooth objective function dependent on a smoothing parameter, this class of methods consists of applying a direct-search algorithm for a fixed value of the smoothing parameter until the step size is relatively small, after which the smoothing parameter is reduced and the process is repeated.

One can show that the worst case complexity (or cost) of this procedure is roughly one order of magnitude worse than the one for direct search or steepest descent on smooth functions.

The class of smoothing direct-search methods is also showed to enjoy asymptotic global convergence properties. Some preliminary numerical experiments indicates that this approach leads to better values of the objective function, pushing in some cases the optimization further, apparently without an additional cost in the number of function evaluations.

Keywords: derivative-free optimization, direct search, smoothing function, worst case complexity, non-smooth, non-convex

1 Introduction

Consider the unconstrained optimization problem

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz continuous function, but not necessarily differentiable or convex. However, given our objective function f we will assume the existence and knowledge of a smoothing function (see [10, 36]):

*Department of Mathematics, University of Coimbra, 3001-454 Coimbra, Portugal (nima@mat.uc.pt). Support for this author was provided by FCT under the scholarship SFRH/BD/33367/2008.

†CMUC, Department of Mathematics, University of Coimbra, 3001-454 Coimbra, Portugal (lnv@mat.uc.pt). Support for this research was provided by FCT under the grant PTDC/MAT/098214/2008.

Definition 1.1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function. We call $\tilde{f} : \mathbb{R}^n \times [0, +\infty) \to \mathbb{R}$ a smoothing function of f if, for any $\mu \in (0, +\infty)$, $\tilde{f}(\cdot, \mu)$ is continuously differentiable in $\mathbb{R}^n$ and, for any $x \in \mathbb{R}^n$,*

$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) \;=\; f(x). \tag{1}$$

Under reasonable assumptions (including boundedness of the objective function level sets), the smoothing direct-search methods derived in this paper will generate a sequence of points (converging to a point x_*) and a sequence of smoothing parameters (converging to zero) for which the gradient of the smoothing function tends to zero. In other words, we will show that x_* is a stationary point of the smoothing function $\tilde{f}$, in the sense that $0 \in G_{\tilde{f}}(x_*)$, with

$$G_{\tilde{f}}(x_*) \;=\; \{v : \exists N \in \mathcal{N}_\infty, (x, \mu) \underset{N}{\longrightarrow} (x_*, 0) \;\text{ with }\; \nabla \tilde{f}(x, \mu) \underset{N}{\longrightarrow} v\}, \tag{2}$$

where $\mathcal{N}_\infty$ represents the set of infinite sequences. As we will see later (see Sections 7 and 8), it is known that for certain types of objective functions and corresponding smoothing functions, $G_{\tilde{f}}(x_*) \subseteq \partial f(x_*)$, where $\partial f(x_*)$ denotes the Clarke subdifferential of f at x_*. Thus, in those cases, the smoothing direct-search methods are capable to generate a sequence of iterates converging to a Clarke stationary point.

A smoothing direct-search method consists of the application of a direct-search method to the function $\tilde{f}(\cdot, \mu)$ for decreasing values of μ. In each outer iteration the value of μ is fixed and a certain number of direct-search inner iterations are applied until the step size becomes lower than a power of μ. Given that $\tilde{f}(\cdot, \mu)$ is continuously differentiable, the choice of direct-search iteration can be simply reduced to polling around the current iterate using a positive spanning set (a set of directions generating $\mathbb{R}^n$ with non-negative coefficients). Such a smoothing direct search is shown to be globally convergent in the sense that for any starting point it generates a subsequence of outer iterates (the output of outer iterations) converging to a point x_* for which $0 \in G_{\tilde{f}}(x_*)$. Furthermore, one can prove also that such a smoothing direct-search method takes a number of iterations and function evaluations one order of magnitude higher in the worst case than steepest descent or direct search on smooth functions (see [34]).

Our work applies for instance to composite functions of the type $f = h(F)$, where h is non-smooth (and for which a smoothing function is known) and F is a vectorial function assumed smooth (continuously differentiable) but in practice given as a black box or zero order oracle (meaning that only function values can be evaluated). Given the popularity of optimization problems in parameter identification and inversion, our approach finds room whenever the fitting is measured in terms of a non-differentiable norm (for which smoothing functions are available) and the underlying simulation does not provide derivatives.

The paper is organized as follows. In Section 2, we review the global convergence and worst case complexity properties of direct-search methods based on a sufficient decrease condition for smooth functions. Then, in Section 3, we review some basic concepts in non-smooth calculus, and explain briefly what are the difficulties in deriving a worst case complexity bound for non-smooth functions. The smoothing direct-search approach is introduced in Section 4 and the corresponding global convergence and worst case complexity properties are described, respectively, in Sections 5 and 6. The paper continues then by reviewing the known properties of smoothing functions of relevance to our work in direct search (the general case in Section 7 and the composite type functions mentioned above in Section 8). A summary of our numerical

experiments with smoothing functions in the context of direct search applied to $f = h(F)$ with $h = \|\cdot\|_1$ is reported in Section 9, and the paper is finished in Section 10 with some concluding remarks.

The notation $\mathcal{O}(M)$ means a multiple of M, where the constant multiplying M does not depend on the dimension n of the problem or on the iteration counter k of the method under analysis (thus depending only on f or on algorithmic constants that are set at the initialization of the method).

2 Direct search and its complexity for smooth functions

Direct-search methods are derivative-free methods which evaluate the objective function at a finite number of points at each iteration and choose the next iterate based on comparisons among such function values. In this paper, we will deal only with direct-search methods of directional type, where descent is achieved along directions belonging to positive spanning sets (see the surveys in [13, 25]).

Each iteration of these direct-search algorithms consists of a search step and a poll step. The search step is optional and irrelevant from the point of view of proving global convergence. Its goal is strictly to improve the numerical performance of the method. When a search step is not performed or is performed unsuccessfully (see Algorithm 2.1 below), a poll step is taken around the current iterate by the means of a step size parameter α_j and a positive spanning set D_j. The poll step evaluates the objective function at the poll points in $P_j = \{y_j + \alpha_j d : d \in D_j\}$, being successful if it can sufficiently reduce the value of the objective function at y_j.

Given the (first order) continuously differentiable nature of the objective function, one could work with a single positive spanning set throughout all the iterations, but the analysis of both global convergence [25] and worst case complexity [34] (in the case where a sufficient decrease condition is imposed to accept new iterates) tells us that one can have considerable freedom when choosing the positive spanning sets for polling. In fact, based on the notion of the cosine measure of a positive spanning set D_j (with nonzero vectors), defined by (see [25])

$$\mathrm{cm}(D_j) \;=\; \min_{0 \neq v \in \mathbb{R}^n} \max_{d \in D_j} \frac{v^\top d}{\|v\|\|d\|},$$

the positive spanning sets used by the algorithm can be selected as follows.

Assumption 2.1 *For all j, the positive spanning set D_j used for polling must satisfy $\mathrm{cm}(D_j) \geq \mathrm{cm}_{min}$ and $d_{min} \leq \|d\| \leq d_{max}$ for all $d \in D_j$ (where $\mathrm{cm}_{min} > 0$ and $0 < d_{min} < d_{max}$ are constants).*

A positive spanning set (with nonzero vectors) has a positive cosine measure. One can thus see that Assumption 2.1 allows plenty of freedom to choose positive spanning sets as long as they do not deteriorate significantly (in the sense of becoming close to loosing the positive spanning property).

We are almost ready to present the algorithmic framework (also used in [34]) for the direct-search optimization of smooth functions based on a sufficient decrease condition. As we can see below in Algorithm 2.1, sufficient decrease is imposed in both the search and the poll steps by means of a forcing function [25] (a non-decreasing function $\rho : (0, +\infty) \to (0, +\infty)$ satisfying $\rho(t)/t \to 0$ when $t \downarrow 0$). The most typical examples of forcing functions are $\rho(t) = c\, t^p$, for $p > 1$, $c > 0$.

Algorithm 2.1 (Direct-search method (search-poll based, for smooth functions))

Initialization

Choose y_0 with $f(y_0) < +\infty$, $\alpha_0 > 0$, $0 < \beta_1 \leq \beta_2 < 1$, and $\gamma \geq 1$.

For $j = 0, 1, 2, \ldots$

1. **Search step:** Try to compute a point with $f(y) < f(y_j) - \rho(\alpha_j)$ by evaluating the function f at a finite number of points. If such a point is found, then set $y_{j+1} = y$, declare the iteration and the search step successful, and skip the poll step.
2. **Poll step:** Choose a positive spanning set D_j. Order the set of poll points $P_j = \{y_j + \alpha_j d : d \in D_j\}$. Start evaluating f at the poll points following the chosen order. If a poll point $y_j + \alpha_j d_j$ is found such that $f(y_j + \alpha_j d_j) < f(y_j) - \rho(\alpha_j)$, then stop polling, set $y_{j+1} = y_j + \alpha_j d_j$, and declare the iteration and the poll step successful. Otherwise declare the iteration (and the poll step) unsuccessful and set $y_{j+1} = y_j$.
3. **Mesh parameter update:** If the iteration was successful, then maintain or increase the step size parameter: $\alpha_{j+1} \in [\alpha_j, \gamma\alpha_j]$. Otherwise decrease the step size parameter: $\alpha_{j+1} \in [\beta_1\alpha_j, \beta_2\alpha_j]$.

Imposing sufficient decrease in both the search and the poll steps results in new iterates always satisfying $f(y_{j+1}) < f(y_j) - \rho(\alpha_j)$. A simple consequence of this fact is that the step size α_j will approach zero when the objective function is bounded below. Consider thus the following assumption.

Assumption 2.2 *The function f is bounded below in $L(y_0) = \{y \in \mathbb{R}^n : f(y) \leq f(y_0)\}$.*

The result motivated above can then be formalized as follows (see [13, 25]).

Theorem 2.1 *Let Assumption 2.2 hold. There exists a subsequence J of unsuccessful iterations such that*

$$\lim_{j \in J} \alpha_j = 0.$$

If $L(y_0)$ is bounded, then there exist a point y_* and subsequence J of unsuccessful iterations such that $\lim_{j\in J} \alpha_j = 0$ and $\lim_{j\in J} y_j = y_*$.

We will describe both the global convergence and the worst case complexity results when sufficient decrease is imposed. In the continuously differentiable case, such results are heavily based on the following result (which is taken from [25]; see also [13, Theorem 2.8 and Equation (7.14)]), describing the relationship between the size of the gradient and the step size parameter at unsuccessful iterations.

Theorem 2.2 *Let D_j be a positive spanning set and $\alpha_j > 0$ be given. Assume that ∇f is Lipschitz continuous (with constant $L_{\nabla f} > 0$) in an open set containing all the poll points in P_j. If $f(y_j) \leq f(y_j + \alpha_j d)$, for all $d \in D_j$, then*

$$\|\nabla f(y_j)\| \leq \mathrm{cm}(D_j)^{-1} \left(\frac{L_{\nabla f}}{2} \alpha_j \max_{d \in D_j} \|d\| + \frac{\rho(\alpha_j)}{\alpha_j \min_{d \in D_j} \|d\|} \right). \tag{3}$$

As observed originally in [25], global convergence can be immediately derived from a direct combination of Theorems 2.1 and 2.2.

Theorem 2.3 *Let Assumptions 2.1 and 2.2 hold. Assume also that f is continuously differentiable with Lipschitz continuous gradient on an open set containing $L(y_0)$ (with constant $L_{\nabla f} > 0$). Then, there exists a subsequence J of unsuccessful iterations such that $\lim_{j\in J} \alpha_j = 0$ and*

$$\lim_{j\in J} \nabla f(y_j) \;=\; 0.$$

If $L(y_0)$ is bounded, then there exists a point y_* such that $\nabla f(y_*) = 0$.

From a worst case complexity or cost point of view, one asks the question of, given $\epsilon \in (0,1)$, how many iterations j_1 are needed to reach $\|\nabla f(y_{j_1+1})\| \leq \epsilon$. Such a result was derived by Vicente [34] and is summarized below for an algorithm of the type of Algorithm 2.1 where (i) the search step is either empty or, when applied, uses a number of function evaluations less than the maximum number of function evaluations made in a poll step, and (ii) the positive spanning sets used for polling have a cardinal of the order of n and a cosine measure of the order of $1/\sqrt{n}$ (like, for instance, $D_j = [I \; -I]$ as in coordinate search).

Theorem 2.4 *Consider the application of Algorithm 2.1 when $\rho(t) = c\,t^p$, $p > 1$, $c > 0$. Let Assumptions 2.1 and 2.2 hold. Let f be continuously differentiable with Lipschitz continuous gradient on an open set containing $L(y_0)$ (with constant $L_{\nabla f} > 0$).*

Under these assumptions, to reduce the gradient below $\epsilon \in (0,1)$, Algorithm 2.1 takes at most

$$\mathcal{O}\left(\left(\sqrt{n}L_{\nabla f}\right)^{\frac{p}{\min(p-1,1)}}\,\epsilon^{-\frac{p}{\min(p-1,1)}}\right)$$

iterations, and at most

$$\mathcal{O}\left(n\left(\sqrt{n}L_{\nabla f}\right)^{\frac{p}{\min(p-1,1)}}\,\epsilon^{-\frac{p}{\min(p-1,1)}}\right)$$

function evaluations.

When $p = 2$, these numbers are of $\mathcal{O}\left(nL_{\nabla f}^2\,\epsilon^{-2}\right)$ and $\mathcal{O}\left(n^2L_{\nabla f}^2\,\epsilon^{-2}\right)$, respectively.

The constant in $\mathcal{O}(\cdot)$ depends only on d_{min}, d_{max}, c, p, β_1, β_2, γ, α_0, and on the lower bound f_{low} of f in $L(y_0)$.

We pause now from the presentation of the material needed for our smoothing direct-search approach to briefly review the recent developments on the study of the worst case complexity in non-linear optimization. Nesterov [29, Page 29] first showed that the steepest descent method for unconstrained optimization takes at most $\mathcal{O}(\epsilon^{-2})$ iterations (or gradient evaluations) to drive the norm of the gradient of the objective function below ϵ. It has been proved by Cartis, Gould, and Toint [7] that the worst case bound $\mathcal{O}(\epsilon^{-2})$ for steepest descent is sharp or tight (see the details in [7]). Gratton, Toint, and co-authors [20, 21] and Cartis, Gould, and Toint [6] proved a similar worst case complexity bound of $\mathcal{O}(\epsilon^{-2})$ for trust-region methods and adaptive cubic overestimation methods, respectively, when these algorithms are based on a Cauchy decrease condition. The worst case complexity bound on the number of iterations was reduced to $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ (in the sense that the negative power of ϵ increases) for the cubic regularization of Newton's method (see Nesterov and Polyak [31]) and for the adaptive cubic overestimation method (see Cartis, Gould, and Toint [6]).

Regarding the unconstrained minimization of non-smooth functions, Cartis, Gould, and Toint [8] showed also that the bound $\mathcal{O}(\epsilon^{-2})$ can be retained by some first order methods for objective functions resulting from the composition of a convex, non-smooth term with a continuously differentiable function.

In the domain of derivative-free optimization, and besides the above described resulted derived by Vicente [34] for direct-search methods based on a sufficient decrease condition when applied to smooth functions, Cartis, Gould, and Toint [9] have investigated the worst case complexity of their adaptive cubic overestimation algorithm when using finite differences to approximate derivatives. They derived the following bound on the number of function evaluations required to drive the norm of the gradient below ϵ

$$\mathcal{O}\left((n^2+5n)\frac{1+|\log(\epsilon)|}{\epsilon^{3/2}}\right).$$

Note that the bound $\mathcal{O}(n^2\epsilon^{-2})$ for direct search is worse in terms of the power of ϵ.

Nesterov [30], using Gaussian smoothing, proposes a random derivative-free approach for which he proves that the number of function evaluations needed for bringing the expected norm of the gradient of the smoothing function of a non-convex non-smooth function below $\epsilon \in (0,1)$ (for a smoothing parameter lower than $\epsilon/(n^{\frac{1}{2}}L_f)$, where L_f is a Lipschitz constant of f) is

$$\mathcal{O}\left(\frac{n(n+4)^2}{\epsilon^3}\right). \tag{4}$$

In the presence of smoothness this bound will be $\mathcal{O}(n\epsilon^{-2})$ (for a smoothing parameter lower than $\epsilon/(nL_{\nabla f})$).

3 Difficulties in the extension to the non-smooth case

Let f be Lipschitz continuous near x. As it is well known [12], the Clarke generalized directional derivative at x along v is defined by

$$f^\circ(x;v) \;=\; \limsup_{\bar{x}\to x \;\; t\downarrow 0} \frac{f(\bar{x}+tv)-f(\bar{x})}{t},$$

and the Clarke subdifferential by

$$\partial f(x) \;=\; \{s\in\mathbb{R}^n : f^\circ(x;v)\geq \langle v,s\rangle,\; \forall v\in\mathbb{R}^n\}.$$

A point x_* is Clarke stationary if $f^\circ(x_*;v)\geq 0\;\forall v\in\mathbb{R}^n$ or, in alternative, if $0\in\partial f(x_*)$. Clarke stationarity is a first order necessary condition for local minimization of locally Lipschitz functions.

The Clarke subdifferential enjoys an equivalent characterization which helps motivating the use of smoothing functions for optimization of non-smooth functions. Let D_f be the subset of $\mathbb{R}^n$ where f is differentiable. Rademacher's Theorem says that the set of all non-differentiable points of a locally Lipschitz continuous function is a set of Lebesgue measure 0. Based on such a result, it is shown in [12, Theorem 2.5.1] that

$$\partial f(x) \;=\; \mathrm{co}\{\lim \nabla f(\bar{x}) :\; \bar{x}\to x,\; \bar{x}\in D_f\},$$

where co represents the convex hull operator. One can now realize how close stationarity of a smoothing function, $0 \in G_f(x_*)$ (see (2)), is to true stationarity, $0 \in \partial f(x_*)$.

It seems difficult to derive a worst case complexity bound for direct search in the non-smooth setting, directly and without the use of smoothing functions. In the non-smooth case, the condition $\|\nabla f(y_{j_1+1})\| \leq \epsilon$ (see the argument before Theorem 2.4) can be replaced by $f^\circ(y_{j_1+1}; v) \geq -\epsilon$ for all normalized directions $v \in \mathbb{R}^n$, where $f^\circ(x; d)$ represents the Clarke generalized directional derivative of f at x along d (which, as mentioned before, is guaranteed to exist if f is Lipschitz continuous near x). The first step in the analysis of worst case complexity would be to extend (3) from the smooth to the non-smooth case. However, ultimately, what we would need to show is that when ℓ is the index of an unsuccessful iteration and $f^\circ(y_\ell; v_\ell) < -\epsilon$ for some normalized direction v_ℓ in $\mathbb{R}^n$, the step size α_ℓ is bounded below by a constant times an appropriate power of ϵ. While attempting to prove such a result one sees a difficulty, given the potentially large distance from v_ℓ to any of the (normalized) polling directions used in iteration ℓ (see the details in [18, Section 4.4]).

The lack of a worst case complexity bound for direct search in the non-smooth setting contrasts with the corresponding availability of global convergence results under the assumption of some form of density of the polling directions in the unit sphere (see [35] for the approach using sufficient decrease and [1, 3] for MADS where a simple decrease is used in accepting new iterates). The reader is also pointed to [25] for an example where a direct-search method using a finite number of polling directions may get stuck and to [2] for an example where under similar assumptions the sequence of iterates might converge to point where 0 is not in the Clarke subdifferential.

4 Smoothing direct-search methods

One way of developing an approach for derivative-free non-smooth optimization for which one can measure the worst case complexity of the algorithms is by means of a smoothing function. The idea is simple and consists of applying directly a direct-search method to the smoothing function with a fixed value of the smoothing parameter μ until a certain precision is achieved, after which the smoothing parameter is reduced and the process repeated. The stopping criterion for each inner direct-search optimization is achieved when the step size gets below a certain function $r(\mu)$ of the smoothing parameter μ, to be selected in advance.

Algorithm 4.1 (Smoothing direct-search method (for non-smooth functions))

Initialization

Choose x_0 with $f(x_0) < +\infty$, $\alpha_0 > 0$, $\mu_0 > 0$ and $\sigma \in (0, 1)$.

For $k = 0, 1, 2, \ldots$

1. **Direct Search for a fixed smoothing parameter:** Apply DS (Algorithm 2.1) to $\tilde{f}(\cdot, \mu_k)$ (starting from $y_{0,k} = x_k$) generating points $y_{0,k}, \ldots, y_{j,k}$ until $\alpha_{j+1,k} < r(\mu_k)$.

2. **Update of the smoothing parameter:** Set $x_{k+1} = y_{j,k}$ and decrease the smoothing parameter: $\mu_{k+1} = \sigma\mu_k$.

A stopping criterion for this algorithm would consist of verifying if μ_k is smaller than a given positive tolerance smaller than μ_0.

5 Global convergence of smoothing direct search

The analysis of global convergence of smoothing direct search (Algorithm 4.1) relies heavily on the one for direct search. The boundedness from below of the smoothing functions is not restrictive since the difference from the smoothing functions to the original one depends continuously on μ for a fixed x (see Definition 1.1), and thus Assumption 5.1 is not more restrictive than an assumption like Assumption 2.2 on the original function.

Assumption 5.1 *For all k: the functions $\tilde{f}(\cdot, \mu_k)$ are bounded below in $L(y_{0,k}) = \{y \in \mathbb{R}^n : \tilde{f}(y, \mu_k) \leq \tilde{f}(y_{0,k}, \mu_k)\}$.*

Theorem 5.1 *Let Assumption 5.1 hold. Then the smoothing parameter goes to zero:*

$$\lim_{k \to \infty} \mu_k \; = \; 0.$$

Proof. For each k, one knows, from Theorem 2.1, that $\liminf_{j \to +\infty} \alpha_{j,k} = 0$. Thus, one always reaches the stopping criterion for every k and μ_k is reduced an infinite number of times. ■

Now, for each k, let j_k be the unsuccessful inner direct-search iteration that achieves the stopping criterion $\alpha_{j_k+1,k} < r(\mu_k)$. After having showed that μ_k converges to zero, one then obtains the property given below for the sequences $\{x_k\}$ and $\{\alpha_{j_k,k}\}$ defined by such j_k's. At this point one needs to select the function $r(\mu)$ such that it tends to zero when $\mu \downarrow 0$.

Theorem 5.2 *Let Assumption 5.1 hold. If $\lim_{\mu \downarrow 0} r(\mu) = 0$, then*

$$\lim_{k \to +\infty} \alpha_{j_k,k} \; = \; 0,$$

and if, in addition, $\{x_k\}$ is bounded, there exists a point x_* and a subsequence $K \subseteq \{j_1, j_2, \ldots\}$ such that $\lim_{k \in K} \alpha_{j_k,k} = 0$ and $\lim_{k \in K} x_k = \lim_{k \in K} y_{j_k,k} = x_*$.

To derive global convergence for smoothing direct search it suffices to use Theorem 2.2, which tells (under Assumption 2.1) us that

$$\|\nabla \tilde{f}(x_k, \mu_k)\| \; \leq \; \mathrm{cm}_{min}^{-1} \left(\frac{L_{\nabla \tilde{f}}(\mu_k)}{2} \alpha_{j_k,k} d_{max} + \frac{\rho(\alpha_{j_k,k})}{\alpha_{j_k,k} d_{min}} \right), \tag{5}$$

where $L_{\nabla \tilde{f}}(\mu_k)$ is the Lipschitz constant of $\nabla \tilde{f}(\cdot, \mu_k)$.

Theorem 5.3 *Let Assumptions 2.1 and 5.1 hold. Assume also that $\tilde{f}$ is a smoothing function for f, where, for all k, $\tilde{f}(\cdot, \mu_k)$ has a Lipschitz continuous gradient on an open set containing $L(y_{0,k})$ with constant $L_{\nabla \tilde{f}}(\mu_k) > 0$. Let x_* (which exists if $\{x_k\}$ is bounded) and K be as in Theorem 5.2.*

Under these conditions, if $\lim_{\mu\downarrow 0} r(\mu) = 0$ *and*

$$\lim_{\mu\downarrow 0} L_{\nabla\tilde{f}}(\mu) r(\mu) \;=\; 0, \tag{6}$$

then

$$\lim_{k\in K} \|\nabla \tilde{f}(x_k, \mu_k)\| \;=\; 0$$

and x_* *is a stationary point associated with the smoothing function* $\tilde{f}$*.*

Proof. From (5), $\beta_1 \alpha_{j_k,k} \leq \alpha_{j_k+1,k}$ (see Step 3 of Algorithm 4.1), and $\alpha_{j_k+1,k} < r(\mu_k)$

$$\|\nabla \tilde{f}(x_k, \mu_k)\| \;\leq\; \mathrm{cm}_{min}^{-1} \left(\frac{L_{\nabla\tilde{f}}(\mu_k)}{2\beta_1} r(\mu_k) d_{max} + \frac{\rho(\alpha_{j_k,k})}{\alpha_{j_k,k} d_{min}} \right).$$

The result then follows from Theorems 5.1 and 5.2, and condition (6). ■

Thus, choosing $r(\cdot)$ appropriately (for instance, $r(\mu) = \mu^2$ when $L_{\nabla\tilde{f}}(\mu) = 1/\mu$), leads to a globally convergent Algorithm 4.1.

6 Worst case complexity of smoothing direct search

We start by first analyzing the worst case complexity of Algorithm 4.1 as it stands.

Theorem 6.1 *Consider the application of Algorithm 4.1 when* $\rho(t) = c_1 t^p$ *and* $r(t) = c_2 t^q$*, with* $p, q > 1$ *and* $c_1, c_2 > 0$*. Let Assumptions 2.1 and 5.1 hold.*

Given any $\xi \in (0, 1)$ *such that* $\xi < \mu_0$*, let* k_1 *be the first iteration such that* $\mu_{k_1+1} \leq \xi$*.*

Under these assumptions, Algorithm 4.1 takes at most $\mathcal{O}\left((-\log(\xi))\xi^{-pq}\right)$ *iterations to reduce the smoothing parameter below* $\xi \in (0, 1)$*, i.e., to have* $\mu_{k_1+1} \leq \xi$*.*

Proof. First let us consider each inner loop of Algorithm 4.1 where direct search is applied for a fixed $\mu_k > \xi$. To bound the number of successful iterations we recall that such a loop is repeated while $\alpha_{j+1,k} \geq r(\mu_k) = c_2 \mu_k^q$. Thus, since $\alpha_{j,k} \geq (1/\gamma)\alpha_{j+1,k}$,

$$\tilde{f}(y_{j,k}, \mu_k) - \tilde{f}(y_{j+1,k}, \mu_k) \;\geq\; c_1 (\alpha_{j,k})^p \;\geq\; c_1 (c_2/\gamma)^p (\mu_k^q)^p,$$

and the number of successful iterations $|S_k|$ is bounded by

$$|S_k| \;\leq\; \frac{\tilde{f}(y_{0,k}, \mu_k) - \tilde{f}_{low,k}}{c_1 (c_2/\gamma)^p \mu_k^{pq}},$$

where $\tilde{f}_{low,k}$ is the lower bound of $\tilde{f}(\cdot, \mu_k)$ in $L(y_{0,k})$. To bound the number $|U_k|$ of unsuccessful iterations, since either $\alpha_{j+1,k} \leq \beta_2 \alpha_{j,k}$ or $\alpha_{j+1,k} \leq \gamma \alpha_{j,k}$, we obtain by induction

$$\alpha_{j,k} \;\leq\; \alpha_{0,k} \gamma^{|S_k|} \beta_2^{|U_k|},$$

which in turn implies from $\log(\beta_2) < 0$

$$|U_k| \;\leq\; -\frac{\log(\gamma)}{\log(\beta_2)} |S_k| - \frac{\log(\alpha_{0,k})}{\log(\beta_2)} + \frac{\log(\alpha_{j,k})}{\log(\beta_2)}.$$

Thus, from $\log(\beta_2) < 0$ and $\alpha_{j,k} \geq r(\mu_k) > r(\xi)$, we conclude that the maximum number of iterations needed in each inner loop minimization is $\mathcal{O}\left(\xi^{-pq}\right)$.

Finally, we need an upper bound for the number of outer loops, i.e., for the number of times that μ is reduced. The smoothing parameter update of Algorithm 4.1 yields $\mu_{k+1} \leq \sigma^k \mu_0$. Hence, in order to have $\mu_{k_1+1} \leq \xi$, we need to have

$$k_1 \ \geq \ \frac{\log(\xi) - \log(\mu_0)}{\log(\sigma)},$$

and the proof is completed ■

Using the best known dependence of $L_{\nabla \tilde{f}}(\mu)$ on μ, which is of the order of $1/\mu$, one can derive the following order of accuracy for the norm of the gradient of the smoothing function (after the effort to reduce μ_k below ξ).

Corollary 6.1 *Consider the application of Algorithm 4.1 when $\rho(t) = c_1 t^p$ and $r(t) = c_2 t^q$, with $p, q > 1$ and $c_1, c_2 > 0$. Let Assumptions 2.1 and 5.1 hold. Assume also that $\tilde{f}$ is a smoothing function for f, where, for all k, $\tilde{f}(\cdot, \mu_k)$ has a Lipschitz continuous gradient on an open set containing $L(y_{0,k})$ with constant $L_{\nabla \tilde{f}}(\mu_k) > 0$ satisfying $L_{\nabla \tilde{f}}(\mu_k) = \mathcal{O}(1/\mu_k)$.*

Given any $\xi \in (0, 1)$ such that $\xi < \mu_0$, let k_1 be the first iteration such that $\mu_{k_1+1} \leq \xi$.

Under these conditions, one has

$$\|\nabla \tilde{f}(x_{k_1}, \mu_{k_1})\| \ = \ \mathcal{O}\left(\xi^{q-1} + \xi^{(p-1)q}\right).$$

Proof. Using Theorem 2.2 (see, rather, (5)) and the fact that $\alpha_{j_{k_1},k_1} < (1/\beta_1) r(\mu_{k_1})$, one can write

$$\|\nabla \tilde{f}(x_{k_1}, \mu_{k_1})\| \ \leq \ \left(\frac{d_{max} c_2}{2\, \mathrm{cm}_{min}\, \beta_1}\right) \{L_{\nabla \tilde{f}}(\mu_{k_1})\} \mu_{k_1}^q + \left(\frac{c_1 c_2^{p-1}}{\mathrm{cm}_{min}\, d_{min} \beta_1^{p-1}}\right) \mu_{k_1}^{(p-1)q}.$$

The proof is completed by nothing that $L_{\nabla \tilde{f}}(\mu_{k_1}) = \mathcal{O}(1/\mu_{k_1})$ and that, from $\mu_{k_1+1} = \sigma \mu_{k_1}$, one has that $\mu_{k_1} \leq \xi/\sigma$. ■

In the following corollary, we will show how to select p and q to achieve the best possible reduction in the norm of the gradient of the smoothing function.

Corollary 6.2 *Under the same assumptions and conditions of Corollary 6.1 and when $q = 2$ and $p = \frac{3}{2}$, Algorithm 4.1 takes at most $\mathcal{O}\left((-\log(\xi))\xi^{-3}\right)$ iterations (and at most $\mathcal{O}\left(n(-\log(\xi))\xi^{-3}\right)$ function evaluations) to reduce the smoothing parameter below $\xi \in (0, 1)$, ending such process with*

$$\|\nabla \tilde{f}(x_{k_1}, \mu_{k_1})\| = \mathcal{O}(n^{\frac{1}{2}} \xi). \tag{7}$$

Proof. The result follows directly from Theorem 6.1 and Corollary 6.1, where the $\mathcal{O}(1/\sqrt{n})$ and $\mathcal{O}(n)$ come, respectively, from the cosine measure and the cardinal of a positive spanning set such as the one used in coordinate search (see Section 2). ■

One can also count the number of iterations and function evaluations needed to reach $\|\nabla \tilde{f}(x_{k_1}, \mu_{k_1})\| \leq \epsilon$ and $\mu_{k_1} \leq \xi = n^{-\frac{1}{2}} \epsilon / C$, where $C > 0$ is the constant that multiplies

$n^{\frac{1}{2}}\xi$ in the right hand side of (7). By replacing ξ in $\mathcal{O}\left((-\log(\xi))\xi^{-3}\right)$ by $n^{-\frac{1}{2}}\epsilon/C$, one obtains the following overall worst case complexity bound in terms of number of iterations

$$\mathcal{O}\left(n^{\frac{3}{2}}[-\log(\epsilon)+\log(n)]\epsilon^{-3}\right),$$

and thus the following overall worst case complexity bound in terms of number of function evaluations

$$\mathcal{O}\left(n^{\frac{5}{2}}[-\log(\epsilon)+\log(n)]\epsilon^{-3}\right). \tag{8}$$

We can compare this worst case complexity bound on function evaluations to the one achieved by Nesterov [30] also for the non-smooth and derivative-free case, given in (4), and conclude that ours is slightly better in terms of the power of n.

Another perspective on the issue of the worst case complexity of Algorithm 4.1 can be taken by considering μ_k fixed and already no larger than a prescribed tolerance ξ and to measure how many iterations would it then be required from direct search to lead to a gradient of the smoothing function smaller than ϵ. In this case, we can apply directly Theorem 2.4 and conclude that the number of iterations required would be

$$\mathcal{O}\left(\left(\sqrt{n}L_{\nabla\tilde{f}}(\mu)\right)^{\frac{p}{\min(p-1,1)}}\epsilon^{-\frac{p}{\min(p-1,1)}}\right).$$

When $L_{\nabla\tilde{f}}(\mu)=\mathcal{O}(1/\mu)$ and $\mu\leq\xi=\epsilon$, one then obtains

$$\mathcal{O}\left((\sqrt{n})^{\frac{p}{\min(p-1,1)}}\epsilon^{-\frac{2p}{\min(p-1,1)}}\right),$$

leading to the optimal choice $p=2$ and a worst case cost of $\mathcal{O}(n\epsilon^{-4})$ direct-search iterations, and thus $\mathcal{O}(n^2\epsilon^{-4})$ in function evaluations. If we choose $\xi=n^{-\frac{1}{2}}\epsilon$, similar as we did in the paragraph above, then these bounds become $\mathcal{O}(n^2\epsilon^{-4})$ and $\mathcal{O}(n^3\epsilon^{-4})$, respectively. It is thus interesting to realize that such a cost is worse than the cost of Algorithm 4.1 (in terms of ϵ, see (8)), suggesting that a strategy where μ is progressively reduced might be advantageous.

Following what Nesterov [29, Page 29] states for first order oracles (see also [34] for the smooth zero order case), one can consider the following problem class (where one can only evaluate the objective function and not its derivatives):

Model:	Unconstrained minimization f Lipschitz continuous f bounded below
Oracle:	Zero order oracle (evaluation of f)
ϵ–solution:	$\mu\leq\epsilon$, $\|\nabla\tilde{f}(x_*^{appr},\mu)\|\leq\epsilon$

where x_*^{appr} is the approximated solution found (given a starting point x_0 for a method). Our result in the paragraph above implies that the number of calls of the oracle is $\mathcal{O}(n^2\epsilon^{-4})$, and thus establishes an upper complexity bound for the above problem class. Moreover, if we redefine our ϵ–solution as $\mu\leq n^{-\frac{1}{2}}\epsilon/C$, $\|\nabla\tilde{f}(x_*^{appr},\mu)\|\leq\epsilon$, where $C>0$ is the constant that multiplies $n^{\frac{1}{2}}\xi$ in the right hand side of (7), the upper complexity bound becomes (8).

We would also like to point out that we have ignored a possible dependence on n in $L_{\nabla\tilde{f}}(\mu)=\mathcal{O}(1/\mu)$. Such dependence might occur or not, as we will see in the next two sections (in the case of Section 8 there is no explicit dependence on n).

7 Construction of smoothing functions

As we have seen, smoothing direct-search methods are capable of generating a sequence of iterates converging to a stationary point associated with the smoothing function. However, it remains to know if such a point is indeed a Clarke stationary point, and thus determining the relationship between the sets $G_{\tilde{f}}(x_*)$ and $\partial f(x_*)$ is of main important for us. We have also seen that is mandatory to make precise the dependence of the Lipschitz constant $L_{\nabla \tilde{f}}(\mu)$ of the gradient of the smoothing function explicitly in terms of the smoothing parameter μ. In this section we will review the results of interest to us on the construction of smoothing functions in the general case, meaning without assuming a specific structure of the function f to be smoothed.

There are various techniques for constructing a smoothing function. One possible way is by convolution with mollifiers [33] (see also [17, 36]). A parameterized family or sequence of measurable functions $\{\psi^\mu : \mathbb{R}^n \to [0, +\infty), \mu \in (0, +\infty)\}$ is called a (bounded) mollifier or mollifier sequence if $\int_{\mathbb{R}^n} \psi^\mu(z)dz = 1$ and if $B^\mu = \{z : \psi^\mu(z) > 0\}$ forms a (bounded) sequence converging to $\{0\}$ as $\mu \downarrow 0$. A smoothing function can be constructed through convolution or averaging with mollifiers [33]:

$$\tilde{f}(x, \mu) \;=\; \int_{\mathbb{R}^n} f(x - z)\psi^\mu(z)dz \;=\; \int_{\mathbb{R}^n} f(z)\psi^\mu(x - z)dz. \tag{9}$$

It is known (see [33, Example 7.19]) that such a sequence of mollifiers converges pointwise to f, i.e., it verifies condition (1) required in the definition of a smoothing function. Furthermore, if the mollifiers $\{\psi^\mu\}$ are continuous on $\mathbb{R}^n$, then the functions $\tilde{f}(\cdot, \mu)$ are continuously differentiable, (this fact is a direct application of [33, Theorem 9.67] in the case where f is Lipschitz continuous near x_*), and thus $\tilde{f}$ is indeed a smoothing function of f. Under these assumptions one also knows from [33, Theorem 9.67] that such $\tilde{f}$ satisfies the gradient consistency property

$$\partial f(x_*) \;=\; \operatorname{co} G_{\tilde{f}}(x_*),$$

thus yielding $G_{\tilde{f}}(x_*) \subseteq \partial f(x_*)$ as desired.

A well known family of mollifiers are the so-called Steklov, defined by setting

$$\psi^\mu(z) = \begin{cases} 1/\mu^n & \text{if } z \in [-\mu/2, \mu/2]^n, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

used in [17, 22] to construct smoothing functions by averaging or convolution. Such smoothing functions were used in [24] to introduce a derivative-free version of the gradient sampling algorithm [5]. In particular, it is known that the resulting smoothing function has the form

$$\tilde{f}(x, \mu) \;=\; \frac{1}{\mu^n} \int_{x_1-\mu/2}^{x_1+\mu/2} dz_1 \ldots \int_{x_n-\mu/2}^{x_n+\mu/2} dz_n f(z).$$

Furthermore, if $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz, then the Steklov smoothing function $\tilde{f}(\cdot, \mu)$ is continuously differentiable, and its gradient is given by (see [17])

$$\begin{aligned} \nabla \tilde{f}(x, \mu) \;=\; & \sum_{i=1}^{n} e_i \frac{1}{\mu^n} \int_{x_1-\mu/2}^{x_1+\mu/2} dz_1 \ldots \int_{x_{i-1}-\mu/2}^{x_{i-1}+\mu/2} dz_{i-1} \int_{x_{i+1}-\mu/2}^{x_{i+1}+\mu/2} dz_{i+1} \ldots \int_{x_n-\mu/2}^{x_n+\mu/2} \\ & dz_n [f(z_1, \ldots, z_{i-1}, x_i + \frac{1}{2}\mu, z_{i+1}, \ldots, z_n) - f(z_1, \ldots, z_{i-1}, x_i - \frac{1}{2}\mu, z_{i+1}, \ldots, z_n)], \end{aligned}$$

where e_i is the i–th column of the identity matrix of order n. One can also show that such a gradient is Lipschitz continuous with Lipschitz constant $2nL_f/\mu$, where L_f is the Lipschitz constant of f (see [22]).

Smoothing functions can also be developed by convolution with probability density functions [17, 30]. In fact, given a density function ϱ (thus satisfying $\int_{\mathbb{R}^n} \varrho(z)dz = 1$ and $\varrho(z) \geq 0$ $\forall z \in \mathbb{R}^n$), one can define $\psi^\mu(z) = \varrho(z/\mu)/\mu^n$ and then apply (9). In the approach [30], a Gaussian density function is used and the gradient of the smoothing function has been proved to be Lipschitz continuous with a constant $\mathcal{O}(1/\mu)$.

A number of optimization algorithms for non-smooth optimization have been developed using some form of smoothing of the non-differentiable objective function. One of the earliest contributions, based on previous work by Ermoliev, was made by Katkovnik [23], in the early 70s. He developed 'random search algorithms' as first order methods using the gradient of an averaged (smoothing) function computed via convolution with mollifiers defined by probability density functions. Several authors further investigated smoothing in optimization, see Kreimer and Rubinstein [26] for a summary of those and a generalization of Katkovnik's work, and the papers [10, 17, 22].

More recently, smoothing techniques have been used in derivative-based optimization by Zhang and Chen [36] to derive a smoothing projected gradient method for non-smooth and nonconvex optimization over a constrained set (involving an application to the solution of stochastic linear complementarity problems where the non-smoothness of the resulting objective function comes from the min operator), and by Chen and Zhou [11] to develop a smoothing nonlinear conjugate gradient method for non-smooth and nonconvex unconstrained optimization problems (solving an application in image restoration where the non-smoothness of the objective function is determined from the absolute value operator). On the derivative-free side, Liuzzi and Lucidi [27] have proposed an algorithm for inequality constrained optimization using a smoothing form of an ℓ_∞ penalty function.

8 A smoothing function for $\|F(\cdot)\|_1$

The ℓ_1 norm is widely used in applications of optimization problems. One possible usage is as an error distance in parameter identification or inverse problems, replacing the role of the ℓ_2 norm in least-squares problems. Another popular role is in sparse optimization and compressed sensing. We are thus interested in defining a smoothing function for the ℓ_1 norm and, more precisely, for a function of the type $\|F(\cdot)\|_1$ where F is a continuously differentiable vectorial operator from $\mathbb{R}^n$ to $\mathbb{R}^m$.

Such a smoothing function will be provided by first pointing out a smoothing function for the absolute value and, later, to use composition of functions and non-smooth calculus rules to pass from $|\cdot|$ to $\|F(\cdot)\|_1$. Chen and Zhou [11] have introduced the following smoothing function for $|\cdot|$:

$$\tilde{s}(t,\mu) \;=\; \int_{-\infty}^{+\infty} |t - \mu\tau|\varrho(\tau)d\tau, \tag{11}$$

where $\varrho : \mathbb{R}^n \to [0, +\infty)$ is a piecewise continuous density function with a finite number of pieces satisfying

$$\varrho(\tau) \;=\; \varrho(-\tau) \quad \text{and} \quad \int_{-\infty}^{+\infty} |\tau|\varrho(\tau)d\tau \;<\; +\infty.$$

Let $\kappa = \int_{-\infty}^{+\infty} |\tau| \varrho(\tau) d\tau$. The following proposition, which is a special case of [11, Proposition 3.1], describes the relevant properties of $\tilde{s}(t, \mu)$.

Proposition 8.1 *The function $\tilde{s}(t, \mu)$ defined by (11) has the following properties:*
(i) $\tilde{s}(t, \mu) = \tilde{s}(-t, \mu)$ for $t \in \mathbb{R}$, that is, $\tilde{s}(\cdot, \mu)$ is symmetric.
(ii) $\tilde{s}(\cdot, \mu)$ is continuously differentiable on $\mathbb{R}$, and its derivative can be given by

$$\tilde{s}'(t, \mu) \;=\; 2 \int_0^{\frac{t}{\mu}} \varrho(\tau) d\tau.$$

(iii) $\tilde{s}(\cdot, \mu)$ converges uniformly to $|t|$ on $\mathbb{R}$ with

$$|\tilde{s}(t, \mu) - |t|| \;\leq\; \kappa \mu.$$

(iv) The set of limits of the derivatives $\tilde{s}'(t, \mu)$ coincides with the Clarke subdifferential of the absolute value, that is,

$$\left\{ \lim_{t \to 0, \mu \downarrow 0} \tilde{s}'(t, \mu) \right\} \;=\; [-1, 1] \;=\; \partial |\cdot|(0) \quad \textit{and} \quad \lim_{t \to t_*, \mu \downarrow 0} \tilde{s}'(t, \mu) \;=\; \begin{cases} 1 & t_* > 0, \\ -1 & t_* < 0. \end{cases}$$

Moreover, one has

$$\lim_{\mu \downarrow 0} \tilde{s}'(t, \mu) \;=\; \begin{cases} 1 & t > 0, \\ 0 & t = 0, \\ -1 & t < 0. \end{cases}$$

(v) For any fixed $\mu > 0$, $\tilde{s}'(t, \mu)$ is Lipschitz continuous with constant $2\kappa_0/\mu$, where κ_0 is an upper bound for ϱ.

If one considers the following uniform density function [11],

$$\varrho(\tau) \;=\; \begin{cases} 1 & \text{if } \tau \in [-\frac{1}{2}, \frac{1}{2}], \\ 0 & \text{otherwise,} \end{cases}$$

then, using (11), the smoothing function for $|\cdot|$ corresponding to this density function is

$$\tilde{s}(t, \mu) \;=\; \begin{cases} \frac{t^2}{\mu} + \frac{\mu}{4} & \text{if } t \in [-\frac{\mu}{2}, \frac{\mu}{2}], \\ |t| & \text{otherwise,} \end{cases}$$

with gradient given by

$$\tilde{s}'(t, \mu) \;=\; \begin{cases} \frac{2t}{\mu} & \text{if } t \in [-\frac{\mu}{2}, \frac{\mu}{2}], \\ \operatorname{sign}(t) & \text{otherwise.} \end{cases}$$

The Lipschitz constant of $\tilde{s}'(\cdot, \mu)$ is $2/\mu$. Note that this smoothing function is precisely the one that is derived using the Steklov mollifier (10).

As mentioned in the beginning of this section, we are interested in the minimization of the ℓ_1 norm of a vectorial function $F : \mathbb{R}^n \to \mathbb{R}^m$, where $F(x) = (F_1(x), \ldots, F_m(x))^\top$ and each F_i is a continuous differentiable function (with a Lipschitz continuous gradient), and we are interested in doing so by means of the smoothing direct-search approach suggested in this paper. Thus,

we are looking for a smoothing function of $\|F(\cdot)\|_1$. The possibility we will explore in this paper is based on the above given smoothing function for $|\cdot|$ and is given by

$$\tilde{F}(x,\mu) \;=\; \sum_{i=1}^{m} \tilde{s}(F_i(x),\mu). \tag{12}$$

We will show that $\tilde{F}$ is indeed a smoothing function for $\|F(\cdot)\|_1$, satisfying the gradient consistency property and exhibiting a Lipschitz continuous gradient with constant of the order of $1/\mu$. Such properties will result from the corresponding properties of $\tilde{s}$ and the use of non-smooth calculus rules for regular functions.

In fact, the absolute value is a regular function, as defined in [12]. A function $g : \mathbb{R}^n \to \mathbb{R}$ is said to be regular at x if it is Lipschitz continuous near x and, for all v, the traditional one-sided directional derivative $g'(x;v)$ exists and coincides with $g^\circ(x;v)$. For instance, all Lipschitz continuous convex functions and continuous differentiable functions are regular [12, Proposition 2.3.6]. The following lemma describes the relations of interest to us between the Clarke subdifferential of sum and composition of functions and the corresponding individual subdifferentials (for a proof see [12, Theorem 2.3.9]).

Theorem 8.1 *Consider a function like $h(F(x))$, where $F : \mathbb{R}^n \to \mathbb{R}^m$ is Lipschitz continuous near x and $h : \mathbb{R}^m \to \mathbb{R}$ is Lipschitz continuous near $F(x)$. Then $h(F)$ is Lipschitz continuous near x and*

$$\partial(h(F))(x) \;\subseteq\; \overline{\text{co}}\left\{\sum_{i=1}^{m} \alpha_i\zeta_i :\; \zeta_i \in \partial F_i(x),\; \alpha = (\alpha_1,\ldots,\alpha_m)^\top \in \partial h(F(x))\right\},$$

where $\overline{\text{co}}$ denotes the closed convex hull. If h is regular at $F(x)$ and F is continuously differentiable at x, $\overline{\text{co}}$ is superfluous and equality holds.

We are now ready to show the desired properties about our smoothing function (12).

Theorem 8.2 *Let $\tilde{F}(x,\mu) = \sum_{i=1}^m \tilde{s}(F_i(x),\mu)$ be defined as in (12). Then*
(i) $\tilde{F}$ is a smoothing function for $\|F\|_1$.
(ii) $\tilde{F}(\cdot,\mu)$ satisfies the gradient consistent property, that is,

$$\left\{\lim_{x\to x_*,\mu\downarrow 0} \nabla\tilde{F}(x,\mu)\right\} \;=\; \partial\|F\|_1(x_*).$$

(iii) For each μ, $\nabla\tilde{F}(\cdot,\mu)$ is Lipschitz continuous with a Lipschitz constant of the order of $1/\mu$.

Proof. (i) From Proposition 8.1.iii and the smoothness of F one can easily show that

$$\lim_{z\to x_*,\mu\downarrow 0} \tilde{F}(z,\mu) \;=\; \|F(x)\|_1.$$

Furthermore, from Proposition 8.1.ii and the smoothness of F, we obtain continuous gradients for $\tilde{F}(\cdot,\mu)$

$$\nabla\tilde{F}(z,\mu) \;=\; \sum_{i=1}^{m} \tilde{s}'(F_i(x),\mu)\nabla F_i(x),$$

and thus $\tilde{F}$ is a smoothing function of $\|F\|_1$.

(ii) One has the following derivation

$$\begin{aligned}
\left\{\lim_{x\to x_*,\mu\downarrow 0} \nabla\tilde{F}(x,\mu)\right\} &= \left\{\lim_{x\to x_*,\mu\downarrow 0} \sum_{i=1}^{m} \tilde{s}'(F_i(x),\mu)\nabla F_i(x)\right\} \\
&= \left\{\sum_{i=1}^{m} \lim_{x\to x_*,\mu\downarrow 0} \tilde{s}'(F_i(x),\mu)\nabla F_i(x)\right\} \\
&= \left\{\sum_{i=1}^{m} \nabla F_i(x_*) \lim_{x\to x_*,\mu\downarrow 0} \tilde{s}'(F_i(x),\mu)\right\} \\
&= \sum_{i=1}^{m} \nabla F_i(x_*)\partial|\cdot|(F_i(x_*)) \\
&= \partial\|F\|_1(x_*),
\end{aligned}$$

where the last equality is justified by Theorem 8.1 and the penultimate one by Proposition 8.1.iv.

(iii) Since, from Proposition 8.1.v, $\tilde{s}'(\cdot,\mu)$ is Lipschitz continuous, one can easily derive

$$\begin{aligned}
&\|\nabla\tilde{F}(x,\mu) - \nabla\tilde{F}(y,\mu)\| \le \sum_{i=1}^{m} \|\tilde{s}'(F_i(x),\mu)\nabla F_i(x) - \tilde{s}'(F_i(y),\mu)\nabla F_i(y)\| \\
&= \sum_{i=1}^{m} \|\tilde{s}'(F_i(x),\mu)\nabla F_i(x) - \tilde{s}'(F_i(x),\mu)\nabla F_i(y) + \tilde{s}'(F_i(x),\mu)\nabla F_i(y) - \tilde{s}'(F_i(y),\mu)\nabla F_i(y)\| \\
&\le \sum_{i=1}^{m} \left\{|\tilde{s}'(F_i(x),\mu)|\|\nabla F_i(x) - \nabla F_i(y)\| + |\tilde{s}'(F_i(x),\mu) - \tilde{s}'(F_i(y),\mu)|\|\nabla F_i(y)\|\right\} \\
&\le \sum_{i=1}^{m} \left\{|\tilde{s}'(F_i(x),\mu)|L_{\nabla F_i}\|x-y\| + L_{\tilde{s}'}(\mu)|F_i(x) - F_i(y)|\|\nabla F_i(y)\|\right\} \\
&\le \sum_{i=1}^{m} (M_i L_{\nabla F_i} + L_{\tilde{s}'}(\mu)L_{F_i}N_i)\|x-y\|,
\end{aligned}$$

where $L_{\nabla F_i}$, L_{F_i}, and $L_{\tilde{s}'}(\mu)$ are the Lipschitz constants of respectively ∇F_i, F_i, and $\tilde{s}'(\cdot,\mu)$, and M_i and N_i are upper bounds of respectively $|\tilde{s}'(F_i,\mu)|$ and $\|\nabla F_i\|$ in $\mathbb{R}^n$ or in a certain subdomain or level set. ■

In summary, we have identified a continuously differentiable smoothing function for $f(\cdot) = \|F(\cdot)\|_1$, satisfying $G_{\tilde{f}}(x_*) = \partial f(x_*)$ and for which the gradient is Lipschitz continuous with constant $\mathcal{O}(1/\mu)$.

9 Numerical results

We have made a number of experiments regarding the use of smoothing in direct search, on a test set suggested in [28] consisting of 53 problems of the form $\min_{x\in\mathbb{R}^n} f(x) = \|F(x)\|_1$, where F varies among 22 nonlinear vector functions of the CUTEr collection [19] with $2 \le n \le 12$ and different initial points were considered. We chose `sid-psm` as our direct-search solver. This

code performed relatively well in a number of benchmarkings [14, 32], especially among direct-search solvers, in particular due to a search step where a quadratic model is fitted using previous function evaluations and minimized in a trust region, and due to a poll step where polling points are ordered for evaluation using a negative simplex gradient.

We will report here only a limited number of the tests performed. We compared the default version of `sid-psm` to Algorithm 4.1 where we also applied `sid-psm` for each value of μ_k, using the smoothing function (12) for $f(x) = \|F(x)\|_1$. Algorithm 4.1 was run using $\mu_0 = 10^{-2}$, $r(\mu) = \max(10^{-5}, \mu^2)$ (note that 10^{-5} is the default stopping tolerance of `sid-psm` for the step size parameter), and the update $\mu_{k+1} = \mu_k/10$. The algorithm was stopped when μ_k reaches 10^{-3}, which, given the initial value for μ_0, amounts in doing 2 major iterations ($k = 0, 1$). We refer as `Ssid-psm` to such a version of Algorithm 4.1.

We start by depicting in Figure 1 the best value of $f(x) = \|F(x)\|_1$ obtained by the 2 methods/solvers for all the problems.

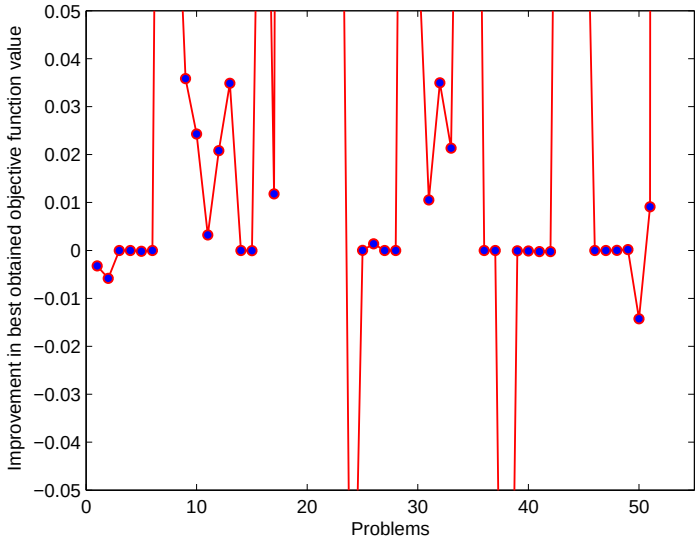

Figure 1: Difference in best function values obtained for a set of piecewise smooth problems. Positive (resp. negative) values indicate better performance of `Ssid-psm`/Algorithm 4.1 (resp. of `sid-psm`).

In Figure 2 we show a data profile [28] indicating the percentage of problems solved by the two solvers under consideration as function of a budget of objective function evaluations (scaled by $n + 1$). A problem is considered solved when

$$f(x_0) - f(x) \;\geq\; (1 - \theta)[f(x_0) - f_L],$$

where $\theta \in (0, 1)$ is a level of accuracy, x_0 is the initial iterate, and f_L is the best objective value found by the two solvers for a budget of 1500 function evaluations. In Figure 2 we set $\theta = 10^{-7}$.

In addition, a performance profile [15] is given in Figure 3, depicting how well a solver performed relatively to the other in reaching the same (scale invariant) convergence test [16], in our case chosen as

$$f(x) - f_* \;\leq\; \theta(|f_*| + 1|),$$

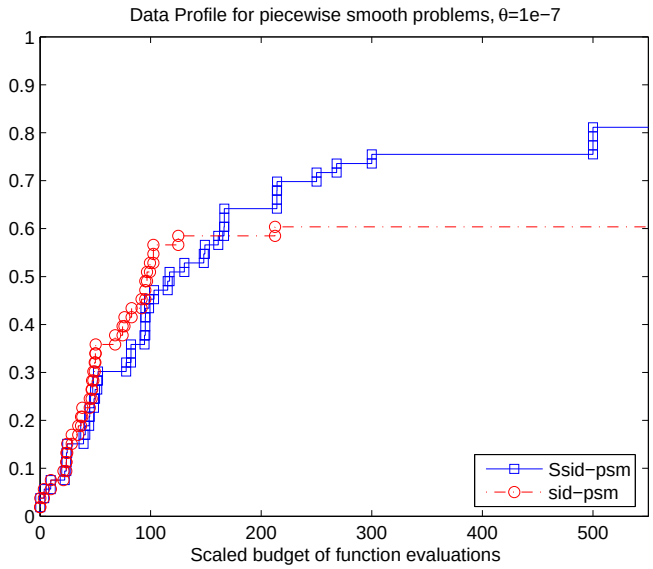

Figure 2: Data profiles computed for a set of piecewise smooth problems. `Ssid-psm` stands for a version of Algorithm 4.1.

where θ is the accuracy level and f_* is an approximation for the optimal value of the problem being tested. For each solver, the respective curve describes (at $\tau = 1$) the fraction of problems for which the solver performs the best (efficiency) and (for τ sufficiently large) the fraction of problems solved by the solver (robustness). In Figure 3 we set $\theta = 10^{-4}$. In an attempt to measure the ability to rigorously solve the problems in hand we set f_* for each problem to the best value attained by these 2 solvers and by those also tested in [14].

Figure 1 indicated that the smoothing direct-search approach led to better objective function values, but did not inform us if such an improvement came at the cost of more function evaluations. The data and performance profiles assured us that indeed one can solve more problems accurately without paying an additional cost in effort. The data profile of Figure 2 tell us that the budget needed to achieve the best precision among the two solvers is approximately the same until a point where `sid-psm` looses in number of problems 'solved'. The performance profile of Figure 3 says that `Ssid-psm` takes roughly less iterations than `sid-psm` in approximately 60% of the problems, loosing in about 30%. In terms of robustness `Ssid-psm` solves approximately 30% times more problems than `sid-psm`, where solving here is now related to (an approximated) optimal value.

Besides varying the accuracy level in data and performance profiles, we also made a number of other tests trying to measure of the impact of smoothing in the search and poll steps separately. For instance we turned off the search step, both in `sid-psm` and in the usage of `sid-psm` in Algorithm 4.1. We then tried the default poll step and another one where the ordering of the polling points is made in a cycling fashion (without the use of a simplex gradient). Since we observed approximately the same order of improvement in using smoothing direct search as before, we will omit the details for sake of brevity. These and other experiments will be reported in the forthcoming PhD thesis of the first author. Overall, we did observe significant gainings in smoothing a function before applying direct search (when such a smoothing function exists,

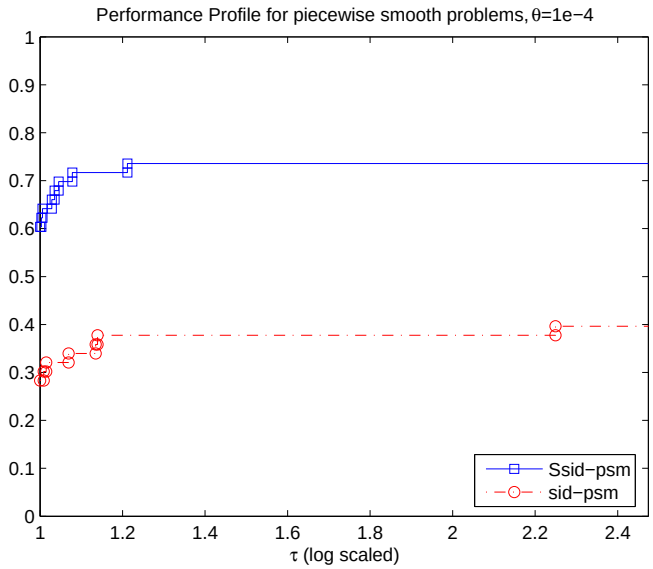

Figure 3: Performance profiles computed a the set of piecewise smooth problems, in a logarithmic scale. `Ssid-psm` stands for a version of Algorithm 4.1.

and for a type of direct search which does not aim at a set of polling directions dense in the unit sphere).

10 Conclusions

Establishing a bound on the worst case complexity or cost of direct-search methods is a nontrivial problem when the function being optimized is non-smooth. In part this is because one needs an infinity of polling directions to approach some form of stationary in the non-smooth case. However, even in the simple situation where the objective function is the sum of a smooth term with a piecewise linear one ($\min_{x\in\mathbb{R}^n} f(x) + \|Ax - b\|_p$, with $p = 1, \infty$, see [4]) for which the number of polling directions can be finite, the authors are incapable of deriving such a bound.

The use of smoothing appears thus as a natural tool to yield the desired bound. As we have seen in our paper, smoothing direct search is not only globally convergent but also exhibits a worst case behavior for which the corresponding number of iterations or function evaluations can be measured. One pays a price in smoothing non-smooth objective functions, in the sense that the bound in the worst case complexity is at least one order worse than the corresponding bound for smooth direct search. This extra order of effort is related to the dependence of the Lipschitz constant of the gradient of the smoothing function on the inverse of the smoothing parameter.

Finally, it should be pointed out that the results of this paper can be extended to bound and linear constraints, where the number of positive generators of the tangent cones of the nearly active constraints is finite.

References

[1] M. A. Abramson and C. Audet. Convergence of mesh adaptive direct search to second-order stationary points. *SIAM J. Optim.*, 17:606–619, 2006.

[2] C. Audet. Convergence results for pattern search algorithms are tight. *Optim. Eng.*, 5:101–122, 2003.

[3] C. Audet and J. E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM J. Optim.*, 17:188–217, 2006.

[4] C. Bogani, M. G. Gasparo, and A. Papini. Generalized set search methods for piecewise smooth problems. *SIAM J. Optim.*, 20:321–335, 2009.

[5] J. V. Burke, A. S. Lewis, and M. L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM J. Optim.*, 15:751–779, 2005.

[6] N. I. M. Gould C. Cartis and Ph. L. Toint. Adaptive cubic overestimation methods for unconstrained optimization. Part II: worst-case function-evaluation complexity. *Math. Program.*, 2012, to appear.

[7] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization. *SIAM J. Optim.*, 20:2833–2852, 2010.

[8] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. Technical Report naXys-06-2011, Département de Mathématiques, FUNDP, Namur (B), 2011.

[9] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the oracle complexity of first-order and derivative-free algorithms for smooth nonconvex minimization. *SIAM J. Optim.*, 22:66–86, 2012.

[10] C. Chen and O. L. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Comput. Optim. Appl.*, 5:97–138, 1996.

[11] X. Chen and W. Zhou. Smoothing nonlinear conjugate gradient method for image restoration using nonsmooth nonconvex minimization. *SIAM J. Imaging Sciences*, 3:765–790, 2010.

[12] F. H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York, (reprinted by SIAM, Philadelphia), 1990.

[13] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[14] A. L. Custódio, H. Rocha, and L. N. Vicente. Incorporating minimum Frobenius norm models in direct search. *Comput. Optim. Appl.*, 46:265–278, 2010.

[15] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.

[16] E. D. Dolan, J. J. Moré, and T. S. Munson. Optimality measures for performance profiles. *SIAM J. Optim.*, 16:891–909, 2006.

[17] Y. M. Ermoliev, V. I. Norkin, and R. J.-B. Wets. The minimization of semicontinuous functions: Mollifier subgradients. *SIAM J. Control Optim.*, 32:149–167, 1995.

[18] R. Garmanjani. *Smoothing and Worst Case Complexity for Direct-Search Methods in Non-Smooth Optimization*. PhD thesis, Dept. Mathematics, Univ. Coimbra, 2012.

[19] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited. 29:373–394, 2003.

[20] S. Gratton, M. Mouffe, Ph. L. Toint, and M. Weber-Mendonca. A recursive trust-region method in infinity norm for bound-constrained nonlinear optimization. *IMA J. Numer. Anal.*, 28:827–861, 2008.

[21] S. Gratton, A. Sartenaer, and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM J. Optim.*, 19:414–444, 2008.

[22] A. M. Gupal. On a method for the minimization of almost-differentiable functions. *Cybernet. Systems Anal.*, 13:115–117, 1977.

[23] V. Y. Katkovnik. Method of averaging operators in iteration algorithms for stochastic optimization. *Cybernet. Systems Anal.*, 4:670–679, 1972.

[24] K. C. Kiwiel. A nonderivative version of the gradient sampling algorithm for nonsmooth nonconvex optimization. *SIAM J. Optim.*, 20:1983–1994, 2010.

[25] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Rev.*, 45:385–482, 2003.

[26] J. Kreimer and R. Y. Rubinstein. Nondifferentiable optimization via smooth approximation: General analytical approach. *Ann. Oper. Res.*, 39:97–119, 1992.

[27] G. Liuzzi and S. Lucidi. A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an ℓ_∞ penalty function. *SIAM J. Optim.*, 20:1–29, 2009.

[28] J. J. Moré and S. M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM J. Optim.*, 20:172–191, 2009.

[29] Y. Nesterov. *Introductory Lectures on Convex Optimization.* Kluwer Academic Publishers, Dordrecht, 2004.

[30] Y. Nesterov. Random gradient-free minimization of convex functions. Technical Report 2011/1, CORE, 2011.

[31] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton's method and its global performance. *Math. Program.*, 108:177–205, 2006.

[32] L. M. Rios and N. Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. 2010.

[33] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis.* Springer, Berlin, 1997, third printing in 2009.

[34] L. N. Vicente. Worst case complexity of direct search. Technical Report 10-17, Dept. Mathematics, Univ. Coimbra, 2010.

[35] L. N. Vicente and A. L. Custódio. Analysis of direct searches for discontinuous functions. *Math. Program.*, 133:299–325, 2012.

[36] C. Zhang and X. Chen. Smoothing projected gradient method and its application to stochastic linear complementarity problems. *SIAM J. Optim.*, 20:627–649, 2009.