Worst Case Complexity of Direct Search

L. N. Vicente*

October 25, 2012

Abstract

In this paper we prove that the broad class of direct-search methods of directional type based on imposing sufficient decrease to accept new iterates shares the worst case complexity bound of steepest descent for the unconstrained minimization of a smooth function, more precisely that the number of iterations needed to reduce the norm of the gradient of the objective function below a certain threshold is at most proportional to the inverse of the threshold squared.

In direct-search methods, the objective function is evaluated, at each iteration, at a finite number of points. No derivatives are required. The action of declaring an iteration successful (moving into a point of lower objective function value) or unsuccessful (staying at the same iterate) is based on objective function value comparisons. Some of these methods are directional in the sense of moving along predefined directions along which the objective function will eventually decrease for sufficiently small step sizes.

The worst case complexity bounds derived measure the maximum number of iterations as well as the maximum number of objective function evaluations required to find a point with a required norm of the gradient of the objective function, and are proved for such directional direct-search methods when a sufficient decrease condition based on the size of the steps is imposed to accept new iterates.

Keywords: derivative-free optimization, direct search, worst case complexity, sufficient decrease

1 Introduction

Direct search is a broad class of methods for optimization without derivatives and includes those of simplicial type [7, Chapter 8], like the Nelder-Mead method and its numerous modifications (where typically one moves away from the worst point), and those of directional type [7, Chapter 7] (where one tries to move along a direction defined by the best or a better point). This paper focuses on direct-search methods of the latter type applied to smooth, continuously differentiable objective functions. The problem under consideration is the unconstrained minimization of a real-valued function, stated as $\min_{x\in\mathbb{R}^n} f(x)$.

Each iteration of the direct-search methods (of directional type) can be organized around a search step (optional) and a poll step, and it is the poll step that is responsible for the

[*] CMUC, Department of Mathematics, University of Coimbra, 3001-454 Coimbra, Portugal (lnv@mat.uc.pt). Support for the author was provided by FCT under the grant PTDC/MAT/098214/2008. This work was developed while the author was visiting the Courant Institute of Mathematical Sciences of New York University under a FLAD scholarship.

global convergence of the overall method (meaning the convergence to some form of stationarity independently of the starting point). In the poll step, the evaluation of the objective function is done at a finite set of polling points defined by a step size parameter and a set of polling directions. When the objective function is continuously differentiable and its gradient is Lipschitz continuous, and for the purpose of global convergence, it suffices that the polling directions form a positive spanning set (i.e., a set of vectors whose linear combinations with non-negative coefficients span $\mathbb{R}^n$), and that new iterates are only accepted when satisfying a sufficient decrease condition based on the step size parameter.

In this paper we will prove that in at most $\mathcal{O}(\epsilon^{-2})$ iterations these methods are capable of driving the norm of the gradient of the objective function below ϵ. It was shown by Nesterov [15, Page 29] that the steepest descent method for unconstrained optimization takes at most $\mathcal{O}(\epsilon^{-2})$ iterations or gradient evaluations for the same purpose (the stepsize is assumed to verify a sufficient decrease condition and another one to avoid too short steps, like a curvature type condition). Direct search-methods that poll using positive spanning sets are directional methods of descent type, and despite not using gradient information as in steepest descent, it is not unreasonable to expect that they share the same worst case complexity bound of the latter method in terms of number of iterations, provided new iterates are only accepted based on a sufficient decrease condition. In fact, it is known that one of the directions of a positive spanning set makes necessarily an acute angle with the negative gradient (when the objective function is continuously differentiable), and, as we will see in the paper, this is what is needed to achieve the same power of ϵ in terms of iterations as of steepest descent. There is an effort in terms of objective function evaluations related to not knowing in advance which of these directions is of descent and to search for the corresponding decrease, but that is reflected in terms of a power of n.

More concretely, based on the properties of positive spanning sets and on the number of objective function evaluations taken in an unsuccessful poll step, we will then conclude that at most $\mathcal{O}(n^2\epsilon^{-2})$ objective function evaluations are required to drive the norm of the gradient of the objective function below ϵ.

As a direct consequence of this result, and following what Nesterov [15, Page 29] states for first order oracles, one can ensure an upper complexity bound for the following problem class (where one can only evaluate the objective function and not its derivatives):

Model:	Unconstrained minimization $f \in C^1_\nu(\mathbb{R}^n)$ f bounded below
Oracle:	Zero order oracle (evaluation of f)
ϵ–solution:	$f(x_*^{appr}) \leq f(x_0)$, $\|\nabla f(x_*^{appr})\| \leq \epsilon$

where f is assumed smooth with Lipschitz continuous gradient (with constant $\nu > 0$), x_*^{appr} is the approximated solution found, and x_0 is the starting point chosen in a method. Our result thus says that the number of calls of the oracle is $\mathcal{O}(n^2\epsilon^{-2})$, and thus establishes an upper complexity bound for the above problem class.

Such an analysis of worst case complexity contributes to a better understanding of the numerical performance of this class of derivative-free optimizations methods. One knows that these methods, when strictly based on polling, although capable of solving most of the problem instances, are typically slow. They are also very appealing for parallel environment due to the natural way of paralelizing the poll step. The global rate of convergence of $n^2\epsilon^{-2}$ indicates that their work (in terms of objective function evaluations) is roughly proportional to the inverse of the square of the targeted accuracy ϵ. It also tells us that such an effort is proportional to the square of the problem dimension n, and this indication is certainly of relevance for computational budgetary considerations.

The structure of the paper is as follows. In Section 2 we describe the class of direct search under consideration. Then, in Section 3, we analyze the worst case complexity or cost of such direct-search methods. Conclusions and extensions of our work are discussed in Section 4. The notation $\mathcal{O}(M)$ in our paper means a multiple of M, where the constant multiplying M does not depend on the iteration counter k of the method under analysis (thus depending only on f or on algorithmic constants set at the initialization of the method). The dependence of M on the dimension n of the problem will be made explicit whenever appropriate. The vector norms will be ℓ_2 ones.

2 Direct-search algorithmic framework

We will follow the algorithmic description of generalized pattern search in [1] (also adopted in [7, Chapter 7] for direct-search methods of directional type). Such framework can describe the main features of pattern search, generalized pattern search (GPS) [1], and generating set search (GSS) [13].

As we said in the introduction, each iteration of the direct-search algorithms under study here is organized around a search step (optional) and a poll step. The evaluation process of the poll step is opportunistic, meaning that one moves to a poll point in $P_k = \{x_k + \alpha_k d : \, d \in D_k\}$, where α_k is a step size parameter and D_k a positive spanning set, once some type of decrease is found.

As in the GSS framework, we include provision to accept new iterates based on a sufficient decrease condition which uses a forcing function. Following the terminology in [13], $\rho : (0, +\infty) \to (0, +\infty)$ will represent a forcing function, i.e., a non-decreasing (continuous) function satisfying $\rho(t)/t \to 0$ when $t \downarrow 0$. Typical examples of forcing functions are $\rho(t) = c\, t^p$, for $p > 1$ and $c > 0$. We are now ready to describe in Algorithm 2.1 the class of methods under analysis in this paper.

Algorithm 2.1 (Directional direct-search method)

Initialization

Choose x_0 with $f(x_0) < +\infty$, $\alpha_0 > 0$, $0 < \beta_1 \leq \beta_2 < 1$, and $\gamma \geq 1$.

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

1. **Search step:** Try to compute a point with $f(x) < f(x_k) - \rho(\alpha_k)$ by evaluating the function f at a finite number of points. If such a point is found, then set $x_{k+1} = x$, declare the iteration and the search step successful, and skip the poll step.
2. **Poll step:** Choose a positive spanning set D_k. Order the set of poll points $P_k = \{x_k + \alpha_k d : \, d \in D_k\}$. Start evaluating f at the poll points following the chosen order.

If a poll point $x_k + \alpha_k d_k$ is found such that $f(x_k + \alpha_k d_k) < f(x_k) - \rho(\alpha_k)$, then stop polling, set $x_{k+1} = x_k + \alpha_k d_k$, and declare the iteration and the poll step successful. Otherwise, declare the iteration (and the poll step) unsuccessful and set $x_{k+1} = x_k$.

3. **Mesh parameter update:** If the iteration was successful, then maintain or increase the step size parameter: $\alpha_{k+1} \in [\alpha_k, \gamma\alpha_k]$. Otherwise, decrease the step size parameter: $\alpha_{k+1} \in [\beta_1\alpha_k, \beta_2\alpha_k]$.

As we will see later, the global convergence of these methods is heavily based on the analysis of the behavior of the step size parameter α_k, in particular on having α_k approaching zero. There are essentially two known ways of enforcing the existence of a subsequence of step size parameters converging to zero in direct search of directional type. One way allows the iterates to be accepted based uniquely on a simple decrease of the objective function but restricts the iterates to discrete sets defined by some integral or rational requirements (see [13, 17]). Another way is to impose a sufficient decrease on the acceptance of new iterates as we did in Algorithm 2.1.

Intuitively speaking, insisting on a sufficient decrease will make the function values decrease by a certain non-negligible amount each time a successful iteration is performed. Thus, under the assumption that the objective function f is bounded from below, it is possible to prove that there exists a subsequence of unsuccessful iterates driving the step size parameter to zero (see [13] or [7, Theorems 7.1 and 7.11 and Corollary 7.2]).

Lemma 2.1 *Let f be bounded from below on $L(x_0) = \{x \in \mathbb{R}^n : f(x) \le f(x_0)\}$. Then Algorithm 2.1 generates an infinite subsequence K of unsuccessful iterates for which $\lim_{k \in K} \alpha_k = 0$.*

One has plenty of freedom to choose the positive spanning sets used for polling when choosing this globalization strategy as long as they do not deteriorate significantly (in the sense of becoming close to loosing the positive spanning property). To quantify such a deterioration, we recall first the cosine measure of a positive spanning set D_k (with nonzero vectors), which is defined by (see [13])

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

A positive spanning set (with nonzero vectors) has a positive cosine measure. Global convergence results essentially from the following fact, which is taken from [8, 13] (see also [7, Theorem 2.4 and Equation (7.14)]) and describes the relationship between the size of the gradient and the step size parameter at unsuccessful iterations.

Theorem 2.1 *Let D_k be a positive spanning set and $\alpha_k > 0$ be given. Assume that ∇f is Lipschitz continuous (with constant $\nu > 0$) in an open set containing all the poll points in P_k. If $f(x_k) \le f(x_k + \alpha_k d) + \rho(\alpha_k)$, for all $d \in D_k$, i.e., the iteration k is unsuccessful, then*

$$\|\nabla f(x_k)\| \;\le\; \mathrm{cm}(D_k)^{-1}\left(\frac{\nu}{2}\alpha_k \max_{d \in D_k}\|d\| + \frac{\rho(\alpha_k)}{\alpha_k \min_{d \in D_k}\|d\|}\right). \tag{1}$$

The positive spanning sets used when globalization is achieved by a sufficient decrease condition are then required to satisfy the following assumption (see [13]), where the cosine measure stays sufficiently positive and the size of the directions does not approach zero or tend to infinite.

Assumption 2.1 *All positive spanning sets D_k used for polling (for all k) must satisfy* $\mathrm{cm}(D_k) \geq \mathrm{cm}_{min}$ *and* $d_{min} \leq \|d\| \leq d_{max}$ *for all* $d \in D_k$ *(where* $\mathrm{cm}_{min} > 0$ *and* $0 < d_{min} < d_{max}$ *are constants).*

One can then easily see from Theorem 2.1 (under Assumption 2.1) that when α_k tends to zero (see Lemma 2.1) so does the gradient of the objective function. Theorem 2.1 will be also used in the next section to measure the worst case complexity of Algorithm 2.1.

3 Worst case complexity

We will now derive the worst case complexity bounds on the number of successful and unsuccessful iterations for direct-search methods in the smooth case (Algorithm 2.1 obeying Assumption 2.1 and using a sufficient decrease condition corresponding to a specific forcing function $\rho(\cdot)$). We will consider the search step either empty or, when applied, using a number of function evaluations not much larger than the maximum number of function evaluations made in a poll step, more precisely we assume that the number of function evaluations made in the search step is at most of the order of n. This issue will be clear later in Corollary 3.2 when we multiply the number of iterations by the number of function evaluations made in each iteration.

As we know, each iteration of Algorithm 2.1 is either successful or unsuccessful. Therefore, in order to derive an upper bound on the total number of iterations, it suffices to derive separately upper bounds on the number of successful and unsuccessful iterations. The following theorem presents an upper bound on the number of successful iterations after the first unsuccessful one (which, from Lemma 2.1, always exists when the objective function is bounded from below). Note that when $p = 2$, one has $p/\min(p-1,1) = 2$.

Theorem 3.1 *Consider the application of Algorithm 2.1 when* $\rho(t) = c\, t^p$, $p > 1$, $c > 0$, *and* D_k *satisfies Assumption 2.1. Let* f *satisfy* $f(x) \geq f_{low}$ *for* x *in* $L(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ *and be continuously differentiable with Lipschitz continuous gradient on an open set containing* $L(x_0)$ *(with constant* $\nu > 0$*).*

Let k_0 *be the index of the first unsuccessful iteration (which must exist from Lemma 2.1). Given any* $\epsilon \in (0,1)$, *assume that* $\|\nabla f(x_{k_0})\| > \epsilon$ *and let* j_1 *be the first iteration after* k_0 *such that* $\|\nabla f(x_{j_1+1})\| \leq \epsilon$. *Then, to achieve* $\|\nabla f(x_{j_1+1})\| \leq \epsilon$, *starting from* k_0, *Algorithm 2.1 takes at most* $|S_{j_1}(k_0)|$ *successful iterations, where*

$$|S_{j_1}(k_0)| \;\leq\; \left\lceil \left(\frac{f(x_{k_0}) - f_{low}}{c\,\beta_1^p L_1^p} \right) \epsilon^{-\frac{p}{\min(p-1,1)}} \right\rceil, \tag{2}$$

with

$$L_1 \;=\; \min\left(1, L_2^{-\frac{1}{\min(p-1,1)}}\right) \quad \text{and} \quad L_2 \;=\; \mathrm{cm}_{min}^{-1}(\nu d_{max}/2 + d_{min}^{-1} c). \tag{3}$$

Proof. Let us assume that $\|\nabla f(x_k)\| > \epsilon$, for $k = k_0, \ldots, j_1$.

If k is the index of an unsuccessful iteration, one has from (1) and Assumption 2.1 that

$$\|\nabla f(x_k)\| \;\leq\; \mathrm{cm}_{min}^{-1} \left(\frac{\nu}{2} d_{max} \alpha_k + d_{min}^{-1} c\, \alpha_k^{p-1} \right),$$

which then implies, when $\alpha_k < 1$,

$$\epsilon \;\leq\; L_2 \alpha_k^{\min(p-1,1)}.$$

If $\alpha_k \geq 1$, then $\alpha_k \geq \epsilon$. So, for any unsuccessful iteration, combining the two cases ($\alpha_k \geq 1$ and $\alpha_k < 1$) and considering that $\epsilon < 1$,

$$\alpha_k \;\geq\; L_1 \epsilon^{\frac{1}{\min(p-1,1)}}.$$

Since at unsuccessful iterations the step size is reduced by a factor of at most β_1 and it is not reduced at successful iterations, one can backtrack from any successful iteration k to the previous unsuccessful iteration k_1 (possibly $k_1 = k_0$), and obtain $\alpha_k \geq \beta_1 \alpha_{k_1}$. Thus, for any $k = k_0, \ldots, j_1$,

$$\alpha_k \;\geq\; \beta_1 L_1 \epsilon^{\frac{1}{\min(p-1,1)}}. \tag{4}$$

Let now $k > k_0$ be the index of a successful iteration. From (4) and by the choice of the forcing function,

$$f(x_k) - f(x_{k+1}) \;\geq\; c\,\alpha_k^p \;\geq\; c\,\beta_1^p L_1^p \epsilon^{\frac{p}{\min(p-1,1)}}.$$

We then obtain, summing up for all successful iterations, that

$$f(x_{k_0}) - f(x_{j_1}) \;\geq\; |S_{j_1}(k_0)| c\,\beta_1^p L_1^p \epsilon^{\frac{p}{\min(p-1,1)}},$$

and the proof is completed. ■

Next, we bound the number of unsuccessful iterations (after the first unsuccessful one).

Theorem 3.2 *Let all the assumptions of Theorem 3.1 hold.*

Let k_0 be the index of the first unsuccessful iteration (which must exist from Lemma 2.1). Given any $\epsilon \in (0,1)$, assume that $\|\nabla f(x_{k_0})\| > \epsilon$ and let j_1 be the first iteration after k_0 such that $\|\nabla f(x_{j_1+1})\| \leq \epsilon$. Then, to achieve $\|\nabla f(x_{j_1+1})\| \leq \epsilon$, starting from k_0, Algorithm 2.1 takes at most $|U_{j_1}(k_0)|$ unsuccessful iterations, where

$$|U_{j_1}(k_0)| \;\leq\; \left\lceil L_3 |S_{j_1}(k_0)| + L_4 + \frac{\log\left(\beta_1 L_1 \epsilon^{\frac{1}{\min(p-1,1)}}\right)}{\log(\beta_2)} \right\rceil,$$

with

$$L_3 \;=\; -\frac{\log(\gamma)}{\log(\beta_2)}, \qquad L_4 \;=\; -\frac{\log(\alpha_{k_0})}{\log(\beta_2)},$$

and L_1 given by (3).

Proof. Since either $\alpha_{k+1} \leq \beta_2 \alpha_k$ or $\alpha_{k+1} \leq \gamma \alpha_k$, we obtain by induction

$$\alpha_{j_1} \;\leq\; \alpha_{k_0} \gamma^{|S_{j_1}(k_0)|} \beta_2^{|U_{j_1}(k_0)|},$$

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

$$|U_{j_1}(k_0)| \;\leq\; -\frac{\log(\gamma)}{\log(\beta_2)} |S_{j_1}(k_0)| - \frac{\log(\alpha_{k_0})}{\log(\beta_2)} + \frac{\log(\alpha_{j_1})}{\log(\beta_2)}.$$

Thus, from $\log(\beta_2) < 0$ and the lower bound (4) on α_k, we obtain the desired result. ■

Using an argument similar as the one applied to bound the number of successful iterations in Theorem 3.1, one can easily show that the number of iterations required to achieve the first unsuccessful one is bounded by

$$\left\lceil \frac{f(x_0) - f_{low}}{c\,\alpha_0^p} \right\rceil .$$

Thus, since $\epsilon \in (0,1)$, one has $1 < \epsilon^{-\frac{p}{\min(p-1,1)}}$ and this number is in turn bounded by

$$\left\lceil \frac{f(x_0) - f_{low}}{c\,\alpha_0^p} \epsilon^{-\frac{p}{\min(p-1,1)}} \right\rceil ,$$

which is of the same order of ϵ as the number of successful and unsuccessful iterations counted in Theorems 3.1 and 3.2, respectively. Combining these two theorems, we can finally state our main result in the following corollary. Note, again, that $p/\min(p-1,1) = 2$ when $p = 2$.

Corollary 3.1 *Let all the assumptions of Theorem 3.1 hold.*

To reduce the gradient below $\epsilon \in (0,1)$, Algorithm 2.1 takes at most

$$\mathcal{O}\left(\epsilon^{-\frac{p}{\min(p-1,1)}}\right) \tag{5}$$

iterations. When $p = 2$, this number is of $\mathcal{O}\left(\epsilon^{-2}\right)$.

The constant in $\mathcal{O}(\cdot)$ depends only on cm_{min}, d_{min}, d_{max}, c, p, β_1, β_2, γ, α_0, on the lower bound f_{low} of f in $L(x_0)$, and on the Lipschitz constant ν of the gradient of f.

Proof. Let j_1 be the first iteration such that $\|\nabla f(y_{j_1+1})\| \leq \epsilon$.

Let k_0 be the index of the first unsuccessful iteration (which must always exist as discussed in Section 2).

If $k_0 < j_1$, then we apply Theorems 3.1 and 3.2, to bound the number of iterations from k_0 to j_1, and the argument above this corollary to bound the number of successful iterations until $k_0 - 1$.

If $k_0 \geq j_1$, then all iterations from 0 to $j_1 - 1$ are successful, and we use the argument above this corollary to bound this number of iterations. ■

Interestingly, the fact that the best power of ϵ is achieved when $p = 2$ seems to corroborate previous numerical experience [18], where different forcing functions of the form $\rho(t) = c\,t^p$ (with $2 \neq p > 1$) were tested but leading to a worse performance.

It is important now to analyze the dependence on the dimension n of the constants multiplying the power of ϵ in (5). In fact, the lower bound cm_{min} on the cosine measure depends on n, and the Lipschitz constant ν might also depend on n. The possible dependence of the Lipschitz constant ν on n will be ignored — global Lipschitz constants appear in all existing worst case complexity bounds for smooth nonconvex optimization, and it is well known that such constants may depend exponentially on the problem dimension n (see also [12]).

The dependence of cm_{min} on n is more critical and cannot be ignored. In fact, we have that $\mathrm{cm}(D) = 1/\sqrt{n}$ when $D = [I \ -I]$ is the positive basis used in coordinate search [13], and $\mathrm{cm}(D) = 1/n$ when D is the positive basis with uniform angles [7, Chapter 2] (by a positive basis it is meant a positive spanning set where no proper subset has the same property). Thus, looking at how the cosine measure appears in (2)–(3) and having in mind the existence of the case $D = [I \ -I]$ (for which $\mathrm{cm}(D) = 1/\sqrt{n}$ and for which the maximum cost of function evaluations per iteration is $2n$), one can state the following result.

Corollary 3.2 *Let all the assumptions of Theorem 3.1 hold. Assume that* $\mathrm{cm}(D_k)$ *is a multiple of* $1/\sqrt{n}$ *and the number of function evaluations per iteration is at most a multiple of* n*.*

To reduce the gradient below $\epsilon \in (0,1)$*, Algorithm 2.1 takes at most*

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

function evaluations. When $p=2$*, this number is* $\mathcal{O}\left(n^2\epsilon^{-2}\right)$*.*

The constant in $\mathcal{O}(\cdot)$ *depends only on* d_{min}*,* d_{max}*,* c*,* p*,* β_1*,* β_2*,* γ*,* α_0*, on the lower bound* f_{low} *of* f *in* $L(x_0)$*, and on the Lipschitz constant* ν *of the gradient of* f*.*

It was shown by Nesterov [15, Page 29] 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. A similar worst case complexity bound of $\mathcal{O}(\epsilon^{-2})$ has been proved by Gratton, Toint, and co-authors [10, 11] for trust-region methods and by Cartis, Gould, and Toint [3] for adaptive cubic overestimation methods, when these algorithms are based on a Cauchy decrease condition. The worst case complexity bound on the number of iterations can be 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 [16]) and for the adaptive cubic overestimation method (see Cartis, Gould, and Toint [3]).

It has been proved by Cartis, Gould, and Toint [4] that the worst case bound $\mathcal{O}(\epsilon^{-2})$ for steepest descent is sharp or tight, in the sense that there exists an example, dependent on an arbitrarily small parameter $\tau > 0$, for which a steepest descent method (with a Goldstein-Armijo line search) requires, for any $\epsilon \in (0,1)$, at least $\mathcal{O}(\epsilon^{-2+\tau})$ iterations to reduce the norm of the gradient below ϵ. The example constructed in [4] was given for $n=1$.

It turns out that in the unidimensional case, a direct-search method of the type given in Algorithm 2.1 (where sufficient decrease is imposed using a forcing function $\rho(\cdot)$) can be cast as a steepest descent method with Goldstein-Armijo line search, when the objective function is monotonically decreasing (which happens to be the case in the example in [4]) and one considers the case $p=2$. In fact, when $n=1$, and up to normalization, there is essentially one positive spanning set with 2 elements, $\{-1,1\}$. Thus, unsuccessful steps are nothing else than reductions of step size along the negative gradient direction. Also, since at unsuccessful iterations (see Theorem 2.1 and Assumption 2.1) one has $\|g_k\| \leq L_2\alpha_k$ where L_2 is the positive constant given in (3) and $g_k = \nabla f(x_k)$, and since successful iterations do not decrease the step size, one obtains $\alpha_k \geq L_1\|g_k\|$ with $L_1 = 1/\max\{L_2,1\} \in (0,1]$. By setting $\gamma_k = \alpha_k/\|g_k\|$, one can then see that successful iterations take the form $x_{k+1} = x_k - \gamma_k g_k$ with $f(x_{k+1}) \leq f(x_k) - c\,\alpha_k^2 \leq f(x_k) - c\,L_1\gamma_k\|g_k\|^2$ (note that if c is chosen in $(0,1)$, we have $c\,L_1 \in (0,1)$).

We make now a final comment (choosing the case $p=2$ for simplicity) on the practicality of the worst case complexity bound derived, given that in derivative-free optimization methods like direct search one does not use the gradient of the objective function. Since as we have said just above $\alpha_k \geq L_1\|\nabla f(x_k)\|$ for all k, we could have stated instead a worst case complexity bound for the number of iterations required to drive α_k below a certain $\epsilon \in (0,1)$ (something now measurable in a computer run of direct search) and from that a bound on the number of iterations required to drive $\|\nabla f(x_k)\|$ below ϵ/L_1. However, we should point out that L_1 depends on the Lipschitz constant ν of ∇f and in practice such approach would suffer from the same unpracticality.

4 Final remarks

The study of worst case complexity of direct search (of directional type) brings new insights about the differences and similarities of the various methods and their theoretical limitations. It was possible to establish a worst case complexity bound for those direct-search methods based on the acceptance of new iterates by a sufficient decrease condition (using a forcing function of the step size parameter) and when applied to smooth functions. Deviation from smoothness (see [2, 18]) poses several difficulties to the derivation of a worst case complexity bound, and such a study will be the subject of a separate paper [9].

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. In this case, it has been shown in [13, 14] that a result similar to Theorem 2.1 can be derived, replacing the gradient of the objective function by

$$\chi(x_k) \;=\; \max_{\substack{x_k+w\in\Omega\\ \|w\|\leq 1}} -\nabla f(x_k)^\top w,$$

where Ω denotes the feasible region defined by the bound or linear constraints. (Note that $\chi(\cdot)$ is a continuous and non-negative measure of stationarity; see [6, Pages 449–451].) Once such a result is at hands, and one uses a sufficient decrease for accepting new iterates, one can show global convergence similarly as in the unconstrained case (see [14, 13]). In terms of worst case complexity, one would also proceed similarly as in the unconstrained case.

Another point we would like to stress is that once we allow new iterates to be accepted based uniquely on a simple decrease of the objective function (together, for globalization purposes, with the restriction that the iterates must lie on discrete sets defined by some integral or rational requirements [13, 17]), the worst case complexity bound on the number of iterations seems only provable under additional strong conditions like the objective function satisfying an appropriate decrease rate. In fact one knows for this class of methods that $\|x_k - x_{k+1}\|$ is larger than a multiple of the stepsize α_k (see [1]). Thus, if $f(x_k) - f(x_{k+1}) \geq \theta\|x_k - x_{k+1}\|^p$ is true for some positive constant θ, then we could proceed similarly as in our paper.

Finally, we would like to mention that the result of our paper has been recently compared in Cartis, Gould, and Toint [5]. These authors derived the following worst case complexity bound (on the number of function evaluations required to drive the norm of the gradient below ϵ, and for the version of their adaptive cubic overestimation algorithm that uses finite differences to compute derivatives)

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

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

Acknowledgments

The author would like to thank Michael L. Overton and Mert Gurbuzbalaban (New York University) for interesting discussions on the topic of the paper.

References

[1] C. Audet and J. E. Dennis Jr. Analysis of generalized pattern searches. *SIAM J. Optim.*, 13:889–903, 2002.

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

[3] N. I. M. Gould C. Cartis and Ph. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: Worst-case function- and derivative-evaluation complexity. *Math. Program.*, 130:295–319, 2011.

[4] 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.

[5] 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.

[6] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods.* MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2000.

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

[8] E. D. Dolan, R. M. Lewis, and V. Torczon. On the local convergence of pattern search. *SIAM J. Optim.*, 14:567–583, 2003.

[9] R. Garmanjani and L. N. Vicente. Smoothing and worst-case complexity for direct-search methods in nonsmooth optimization. *IMA J. Numer. Anal.*, to appear.

[10] 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.

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

[12] F. Jarre. On Nesterov's smooth Chebyshev-Rosenbrock function. *Optim. Methods Softw.*, to appear.

[13] 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.

[14] R. M. Lewis and V. Torczon. Pattern search methods for linearly constrained minimization. *SIAM J. Optim.*, 10:917–941, 2000.

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

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

[17] V. Torczon. On the convergence of pattern search algorithms. *SIAM J. Optim.*, 7:1–25, 1997.

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