Optimization Methods and Software
Vol. 00, No. 00, January 2012, 1–8

A note about the complexity of minimizing Nesterov's smooth Chebyshev-Rosenbrock function

Coralia Cartis[a*], Nicholas I. M. Gould[b] and Philippe L. Toint[c]

[a]*School of Mathematics, University of Edinburgh, The King's Buildings, Edinburgh, EH9 3JZ, Scotland, UK. Email: coralia.cartis@ed.ac.uk*; [b]*Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire, OX11 0QX, England, UK. Email: nick.gould@sftc.ac.uk*; [c]*Namur Center for Complex Systems (naXys) and Department of Mathematics, FUNDP-University of Namur, 61, rue de Bruxelles, B-5000 Namur, Belgium. Email: philippe.toint@fundp.ac.be.*

(*Received August 29, 2011; Revised May 8, 2012*)

This short note considers and resolves the apparent contradiction between known worst-case complexity results for first and second-order methods for solving unconstrained smooth nonconvex optimization problems and a recent note by Jarre (2011) implying a very large lower bound on the number of iterations required to reach the solution's neighbourhood for a specific problem with variable dimension.

Keywords: evaluation complexity, worst-case analysis, nonconvex optimization.

AMS Subject Classification: 90C30, 65K05, 90C60, 68Q25.

1. Introduction

The worst-case complexity of algorithms for unconstrained nonconvex smooth optimization has recently been intensively studied by several authors. In particular, we refer the reader to Vavasis [13], Nesterov [11] and Cartis, Gould and Toint [6] for an analysis of steepest descent, to Gratton, Sartenaer and Toint [8] and Cartis et al. [2, 5] for trust-regions algorithms, to Cartis et al. [6] for Newton's method, to Nesterov and Polyak [12] and Cartis et al. [2, 4, 5] for regularized variants, or to Vicente [14] and Cartis et al. [1] for finite-difference and/or derivative-free schemes. The common feature of all these contributions is that they discuss upper (and sometimes lower) bounds on the number of function evaluations that are necessary for the algorithm under consideration to produce an approximate first-order critical point, that is an iterate at which the Euclidean norm of the objective function's gradient is below some user-prescribed tolerance ϵ. Remarkably, these results show that such bounds have the form

$$\left\lceil \frac{\kappa}{\epsilon^{\alpha}} \right\rceil \tag{1}$$

where κ is a problem-dependent constant and α is an algorithm-dependent constant ranging between 3/2 and 2. These bounds are often sharp [6] and are optimal for

*Corresponding author. Email: coralia.cartis@ed.ac.uk

ISSN: 1055-6788 print/ISSN 1029-4937 online

DOI: 10.1080/1055.6788.YYYY.xxxxxx
http://www.tandfonline.com

some regularization methods [5]. It is important for our purposes to note that κ typically depends, possibly exponentially, on problem dimension via the relevant gradient and perhaps Hessian global Lipschitz constants (which are assumed to exist). We also note that all the algorithms considered in these results are descent methods, in the sense that they generate a sequence of iterates with non-increasing objective function values.

An interesting development occured when F. Jarre recently published a report [10] where he pointed out that, on a specific problem with variable dimension, any descent algorithm would require a number of iterations (and hence of function evaluations) which is exponential in problem dimension to reach the (unique) critical point. Since ϵ and α in (1) are independent of dimension, this behaviour could easily be made compatible with the results mentioned above if the problem's Lipschitz constants depended exponentially on dimension on the domain of interest. However, it turns out that, for the considered example, both these constants depend at most polynomially on the problem size, implying that the bound (1) is also depending sub-exponentially on the problem dimension, and could even be independent of problem dimension. It is the purpose of this short note to resolve this apparent contradiction.

2. Some details

We first need to elaborate on the details of the context. In what follows, we consider the problem

$$\begin{array}{l} \text{minimize } f(x) \\ x \in \mathbb{R}^n \end{array}$$

where f is a twice continuously differentiable possibly nonconvex function from $\mathbb{R}^n$ to $\mathbb{R}$, which is assumed to be bounded below (by some value $f_{\rm low}$). To solve this problem, we may apply the Adaptive Regularization with Cubics (ARC) algorithm; here we focus on the ARC variant that has the best known and optimal worst-case evaluation complexity—the so-called ARC$_{\rm (S)}$ in Cartis et al. [4]—which can be briefly outlined as follows, in the case when approximate Hessians are set to the true Hessian values of f. At iteration k, a step s_k from the current iterate x_k is computed, which approximately minimizes the cubic model

$$m(x_k + s) = \langle g(x_k), s\rangle + \tfrac{1}{2}\langle s, H(x_k)s\rangle + \tfrac{1}{6}\sigma_k\|s\|^3,$$

where $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ are the Euclidean inner product and norm, respectively, where $g(x) \stackrel{\rm def}{=} \nabla_x f(x)$, $H(x) \stackrel{\rm def}{=} \nabla_{xx} f(x)$ and $\sigma_k \geq \sigma_{\min} > 0$ is an adaptive regularization parameter whose value is updated recursively inside the algorithm. In particular, the cubic model is globally minimized over increasing Krylov subspaces that contain the gradient $g(x_k)$ until a suitable termination criterion is satisfied; the latter is a relative error condition on the model's gradient, that is definitely satisfied at any stationary point of the model but seems likely to be achieved sooner (see §4 in Cartis et al. [3] for precise details). The step s_k is accepted and the new iterate x_{k+1} set to x_k+s_k whenever a reasonable fraction of the predicted model decrease $m_k(x_k+s_k)$ is realized by the actual decrease in the objective, $f(x_k)-f(x_k+s_k)$. Since the current weight σ_k has resulted in a successful step, there is no pressing reason to increase it, and indeed there may be benefits in decreasing it if the model overestimates the function locally. Otherwise, we judge that the improvement in objective is

insufficient—indeed there may be no improvement if $f(x_k) \leq f(x_k + s_k)$— and then the step will be rejected and x_{k+1} left as x_k. Under these circumstances, the only recourse available is to increase the weight σ_k prior to the next iteration with the implicit intention of reducing the size of the step. Note that if s_k is chosen as the (exact) global minimizer of $m_k(x_k+s)$ for all $s \in \mathbb{R}^n$ and σ_k is maintained at a sufficiently large value, then this ARC variant is similar to the cubic regularization technique proposed in [12].

Crucially for our purposes, it has been proved [4, 12] that, if we assume that $H(x)$ is Lipschitz continuous (with constant L) on each of the segments $[x_k, x_k + s_k]$ and if we define an ϵ-approximate critical iterate as an iterate x_k such that

$$\|g(x_k)\| \leq \epsilon, \tag{2}$$

where $\epsilon \in (0, 1)$ is a user-specified accuracy, then the ARC algorithm started from the initial point x_0 will produce such an iterate in at most

$$\left\lceil (f(x_0) - f_{\rm low}) \frac{\kappa_{\rm ARC}}{\epsilon^{3/2}} \right\rceil \tag{3}$$

iterations. The constant $\kappa_{\rm ARC}$ only depends (polynomially) on L and (when allowing approximate rather than exact model minimization) also on an upper bound on $\|H(x)\|$ on the segments $[x_k, x_k + s_k]$, as well as on fixed, dimension independent, algorithmic parameters (such as $\sigma_{\min}$). We will also make use of a property of the ARC algorithm, namely that, for all $k \geq 0$, we have

$$\|s_k\| \leq 3 \max \left[\frac{\|H(x_k)\|}{\sigma_k}, \sqrt{\frac{\|g(x_k)\|}{\sigma_k}} \right], \tag{4}$$

(see Lemma 2.2 in [3]).

Jarre's example of slow minimization uses the Chebyshev-Rosenbrock function attributed to Nesterov in [9], which is defined, for some $\rho \geq 1$ and $n \geq 2$, by

$$f(x) = \tfrac{1}{4}(x_1 - 1)^2 + \rho \sum_{i=1}^{n-1} (x_{i+1} - 2x_i^2 + 1)^2 \stackrel{\rm def}{=} \tfrac{1}{4}(x_1 - 1)^2 + \rho \sum_{i=1}^{n-1} v_i(x)^2, \tag{5}$$

and whose gradient is given by

$$g_1(x) = \tfrac{1}{2}(x_1 - 1) - 8\rho x_1 v_1(x) \tag{6}$$

$$g_i(x) = 2\rho[v_{i-1}(x) - 4\, x_i\, v_i(x)], \quad (i = 2, \ldots, n-1), \tag{7}$$

and

$$g_n(x) = 2\rho v_{n-1}(x). \tag{8}$$

The nonzero entries of its Hessian are given (up to symmetry) by

$$H_{1,1}(x) = \tfrac{1}{2} - 8\rho v_1(x) + 32\rho x_1^2, \quad H_{1,2}(x) = -8\rho x_1, \tag{9}$$

$$H_{i,i}(x) = 2\rho(1 - 4v_i(x) + 16x_i^2), \quad H_{i,i+1}(x) = -8\rho x_i, \quad (i = 2, \ldots, n-1) \tag{10}$$

and

$$H_{n,n}(x) = 2\rho, \tag{11}$$

while those of its third derivative tensor $T(x)$ are given by

$$T_{1,1,1}(x) = 96\rho x_1, \quad T_{1,1,2}(x) = -8\rho, \quad T_{1,2,1} = -8\rho, \tag{12}$$

$$T_{i,i,i}(x) = 96\rho x_i, \quad T_{i,i,i+1}(x) = -8\rho, \quad (i = 2, \ldots, n-1). \tag{13}$$

The level contours for this function with $\rho = 400$ are shown in Figure 1, the leftmost graph showing the levels in the (x_1, x_2) plane and the rightmost the levels in the (x_i, x_{i+1}) plane, for any i between 2 and $n-1$. The unique first- (and second-) order critical point is $x_* = (1, 1, \ldots, 1)^T$, which is marked on the upper right of each graph.

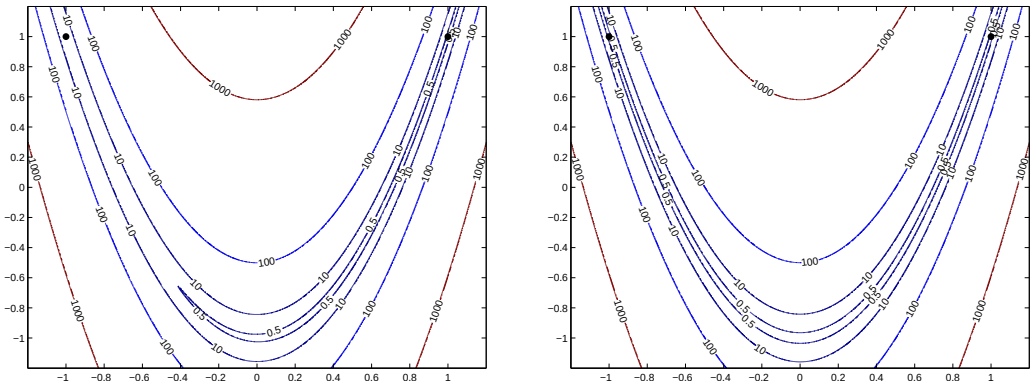

Figure 1. Contours of $f(x)$ with $\rho = 400$ in the (x_1, x_2) plane (left) and in the (x_i, x_{i+1}) plane (for any $2 \leq i \leq n-1$) (right).

The unconstrained minimization of this function is started from $x_0 = (-1, 1, 1, \ldots, 1)^T$ (also marked in the upper left part of the graphs of Figure 1) at which $f(x_0) = 1$ and $\|g(x_0)\| = 1$. Let

$$\mathcal{L}_0 \stackrel{\rm def}{=} \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\},$$

and note that all iterates of any descent algorithm will remain in this level set. It follows from (5) and $f(x_0) = 1$ that any $x = (x_1, \ldots, x_n) \in \mathcal{L}_0$ satisfies

$$\tfrac{1}{2}|x_1 - 1| \leq 1 \quad \text{and} \quad \sqrt{\rho}|x_{i+1} - 2x_i^2 + 1| \leq 1, \quad (i = 1, \ldots, n-1), \tag{14}$$

and so, as $\rho \geq 1$,

$$-1 \leq x_1 \leq 3 \quad \text{and}$$

$$-1 - \frac{1}{\sqrt{\rho}} \leq x_{i+1} \leq 2x_i^2, \quad (i = 1, \ldots, n-1). \tag{15}$$

Thus $x \in \mathcal{L}_0$ is uniformly bounded below independently of n, but may grow (doubly) exponentially as n increases. Indeed, the double-exponential upper bound in

(15) is essentially tight since fixing $\tilde{x}_1 \in (1,3]$ and letting $\tilde{x}_{i+1} = 2\tilde{x}_i^2 - 1$ for $i = 1, \ldots, n-1$, yields $\tilde{x} \in \mathcal{L}_0$ with $\tilde{x}_n$ growing doubly-exponentially with n. In fact, the dependence or otherwise on n of x_n for $x \in \mathcal{L}_0$ entirely determines the amount of growth allowed in the remaining components of x since (14) and $\rho \geq 1$ imply that

$$x_i \leq \sqrt{\tfrac{1}{2}x_{i+1} + 1}, \quad (i = 1, \ldots, n-1), \tag{16}$$

and furthermore, inductively, $x_1 = \mathcal{O}\left(x_n^{1/(2^n)}\right)$. (Since $x_1 \in [-1,3]$, x_n can depend on n at most doubly-exponentially.) In particular, due to (16), if x_n is bounded above independently of n, so are all the other components x_i of $x \in \mathcal{L}_0$. These considerations lead to the following two possible cases, relevant when employing the ARC algorithm.

(1) The component $[x_k]_n$ of all ARC iterates x_k, $k \geq 0$, is uniformly bounded above by a constant depending at most sub-exponentially on n. (This includes the case when $[x_k]_n$ is independent of n.) Then all ARC iterates will remain in $[-\alpha, \alpha]^n$, for some $\alpha > 0$ depending at most sub-exponentially on n. We may therefore derive from (6)–(8) and the sparse nature of (9)–(11) that there exist constants $\kappa_g > 0$ and $\kappa_H > 0$ dependent at most sub-exponentially on n such that, for iterates generated by the ARC algorithm,

$$\|g(x_k)\| \leq \kappa_g \quad \text{and} \quad \|H(x_k)\| \leq \kappa_H \tag{17}$$

for all $k \geq 0$. Moreover, (4) then implies that steps generated by the ARC algorithm satisfy the inequality

$$\|s_k\| \leq 3 \max\left[\frac{\kappa_H}{\sigma_{\min}}, \sqrt{\frac{\kappa_g}{\sigma_{\min}}}\right] \stackrel{\text{def}}{=} \kappa_s.$$

As a consequence, we obtain that, for all $k \geq 0$,

$$[x_k, x_k + s_k] \subset [-\alpha - \kappa_s, \alpha + \kappa_s]^n \stackrel{\text{def}}{=} \mathcal{L}$$

and therefore, using the mean-value theorem, that $H(x)$ is Lipschitz continuous in $\mathcal{L}$ with constant $\max_{x \in \mathcal{L}} \|T(x)\|$, which is itself depending at most sub-exponentially on n, because of the sparsity of T (see (12)–(13)). As a consequence, the value of κ_{ARC} in (3) depends on n at most sub-exponentially, and, because we may obviously choose $f_{\text{low}} = 0$ since $f(x)$ is the sum of squared terms, the upper bound on the maximum number of iterations necessary to achieve (2) starting from x_0 is either fixed or depending at most subexponentially on n, for given ϵ.

On the other hand, Jarre's observation is that when $\rho \geq 400$, any descent algorithm (including ARC) must take

$$\text{at least } \; 1.44 \times 1.618^n \; \text{ iterations} \tag{18}$$

to move from x_0 to x_*, at which $f(x_*) = 0 = f_{\text{low}}$. Moreover, at least half that number of iterations is required to obtain an iterate with $[x_k]_1 \geq 0$, which ensures that (3) cannot be interpreted as an upper bound on the number of iterations needed to reach an ϵ-dependent neighbouhood of x_*.

The next section elucidates this apparent contradiction between (3) and (18).

(2) Some ARC iterates depend (at least) exponentially on n, which is allowed by (14). Then (17) holds with κ_g and κ_H now depending (at least) exponentially on n; similarly, the Lipschitz constant of the Hessian depends (at least) exponentially on n on the path of the iterates. Thus, in this case, the upper bound (3) depends (at least) exponentially on n, and so it is in agreement with the lower bound (18). (Note that even if ARC is initialized with a starting point that depends doubly-exponentially on n, (18) remains consistent with (3).)

Our numerical experiments with ARC applied to function (5) with $\rho \geq 400$ and $x_0 = (-1, 1, \ldots, 1)$ invariably generated iterates with $[x_k]_n \leq 1$. Thus numerically, we can guarantee that we are in Case 1 above, specifically, when (3) is independent of n. However, we have not been able to show analytically that the ARC iterates do not reach the "bad", exponentially dimension-dependent, part of the level set $\mathcal{L}_0$ for the second-order models that we use.

3. Resolving the apparent contradiction

Assume that we are in Case 1 above. We first notice that (3) and (18) are obviously compatible if

$$\epsilon \leq \left(\frac{\kappa_{\rm ARC}}{1.44 \times 1.618^n} \right)^{2/3} \stackrel{\rm def}{=} \theta(n), \tag{19}$$

as in this case the accuracy requirement is tight enough to allow for the number of steps indicated by Jarre's bound. But what happens if (19) is violated is not clear. Using the famous Sherlock Holmes adage that "When you have eliminated the impossible, whatever remains, however improbable, must be the truth" (Conan Doyle, 1890 [7]), we must conclude in this case that, if an ϵ-approximate first-order critical point can be reached in a number of iterations that is either dimension-independent or depending subexponentially on n, but that this point cannot be x_*, then it must be that $f(x)$ admits other *approximate* critical points in $\mathcal{L}_0$ within a fixed or polynomial distance from x_0. And indeed this happens to be the case. The leftmost graph of Figure 2 shows (as a continuous line) the evolution with n of

$$\tau(n) = \min_{x \in \{x_1, \ldots, x_{50}\}} \|g(x)\|,$$

where the x_k are the iterates generated by the ARC algorithm applied to minimize $f(x)$ (with dimension n), starting from x_0. The dashed line in the same graph corresponds to the parallel evolution of $\theta(n)$, the right-hand side of (19). The distance

$$\delta(n) = \left\| x_0 - \arg \min_{x \in \{x_1, \ldots, x_{50}\}} \|g(x)\| \right\|$$

is shown in the rightmost graph.

We may conclude from this figure that, for ϵ above the threshold given by (19), suitable approximate first-order critical points of $f(x)$ exist close to x_0 (and can be found relatively easily by standard optimization methods). A further investigation of these approximate critical points is possible, using the analytical expression of

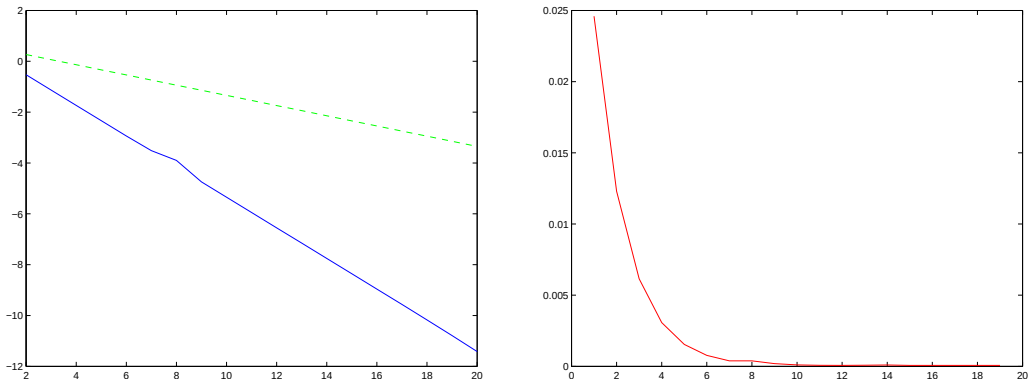

Figure 2. Evalution of $\tau(n)$ and $\theta(n)$ (dashed) (left, in $\log_{10}$ scale) and $\delta(n)$ (right) as functions of n.

$f(x)$. Without entering into too much detail, we may simply say that the gradient norm at such points is dominated by the magnitude of g_n, which is proportional to $|v_{n-1}|$ because of (8). As it turns out, (7) and the fact that all g_i $(i = 2, \ldots, n-1)$ must also be small impose that the $|v_i|$ decrease as an approximate geometric progression. The freedom left for each $|g_i|$ to be small (of the order of $|g_n|$) is enough to counterbalance the effect of x_1 in g_1 given by (6). However, this explanation remains problem specific, which considerably limits its interest and applicability.

Note that a similar apparent contradiction is encountered when trust-region methods are applied to (5). Namely, their associated worst-case complexity of $\mathcal{O}\left(\epsilon^{-2}\right)$ iterations [8] depends at most polynomially on problem size, in apparent contrast to the exponential lower bound (18). However, trust-region methods that we experimented with also generate an approximate local minimizer within 10 iterations.

Similarly, steepest descent methods also satisfy an upper complexity bound $\mathcal{O}\left(\epsilon^{-2}\right)$ that depends linearly on the Lipschitz constant of the gradient ([11], p. 29). Thus, a seeming contradiction with the exponential bound (18) occurs when applying such methods with exact or inexact linesearches to minimizing (5). We expect a similar resolution to the ARC and trust-region cases, namely, the finding of approximate stationary points on the way, before any of these two bounds are achieved. However it will likely be difficult to verify this in practice as steepest descent is often observed to stagnate on Rosenbrock's example ($n = 2$) in floating-point arithmetic.

It remains remarkable that our analysis shows the existence of (potentially many) approximate first-order critical points for a dimension-dependent family of smooth functions for which the gradient and Hessian Lipschitz constants are either dimension independent or depending polynomially on dimension, at a level of approximation which improves exponentially with problem size. It is the authors' view that the implications of this observation (for instance on the geometry of smooth infinite dimensional maps) deserves more study.

Acknowledgements

The work of the first and second author is funded by EPSRC Grant EP/I028854/1 and EP/E053351/1, respectively. All three authors are grateful to the Royal Society for its support through the International Joint Project 14265. The authors are also grateful to Florian Jarre for some interesting comments.

References

[1] 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(1) (2012), pp. 66–86.

[2] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Complexity bounds for second-order optimality in unconstrained optimization*, J. Complexity, 28(1) (2012), pp. 93–108.

[3] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results*, Math. Program. 127(2) (2011), pp. 245–295.

[4] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity*, Math. Program. 130(2) (2011), pp. 295–319.

[5] C. Cartis, N. I. M. Gould, and Ph. L. Toint, *Optimal Newton-type methods for nonconvex optimization*, ERGO Tech. Report 11-009, School of Mathematics, University of Edinburgh, UK, 2011.

[6] 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(6) (2010), pp. 2833–2852.

[7] A. Conan Doyle, *The sign of the four*, Lippincott's Monthly Magazine, February 1890.

[8] S. Gratton, A. Sartenaer, and Ph. L. Toint, *Recursive trust-region methods for multiscale nonlinear optimization*, SIAM J. Optim. 19(1) (2008), pp. 414–444.

[9] M. Gürbüzbalaban and M. L. Overton, *On Nesterov's nonsmooth Chebyshev-Rosenbrock functions*, Nonlinear Analysis 75 (2012), pp. 1282–1289.

[10] F. Jarre, *On Nesterov's smooth Chebyshev-Rosenbrock function*, Optim. Methods Softw. DOI:10.1080/10556788.2011.638924, 2011.

[11] Yu. Nesterov, *Introductory Lectures on Convex Optimization*. Applied Optimization. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[12] Yu. Nesterov and B. T. Polyak, *Cubic regularization of Newton's method and its global performance*, Math. Program. 108(1) (2006), pp. 177–205.

[13] S. A. Vavasis, *Black-box complexity of local minimization*, SIAM J. Optim. 3(1) (1993), pp. 60–80.

[14] L. N. Vicente, *Worst case complexity of direct search*, Preprint 10-17, Department of Mathematics, University of Coimbra, Portugal, 2010.