Globally Convergent Evolution Strategies for Constrained Optimization

Y. Diouane[*]   S. Gratton[†]   L. N. Vicente[‡]

March 17, 2015

Abstract

In this paper we propose, analyze, and test algorithms for constrained optimization when no use of derivatives of the objective function is made. The proposed methodology is built upon the globally convergent evolution strategies previously introduced by the authors for unconstrained optimization. Two approaches are encompassed to handle the constraints. In a first approach, feasibility is first enforced by a barrier function and the objective function is then evaluated directly at the feasible generated points. A second approach projects first all the generated points onto the feasible domain before evaluating the objective function.

The resulting algorithms enjoy favorable global convergence properties (convergence to stationarity from arbitrary starting points), regardless of the linearity of the constraints.

The algorithmic implementation (i) includes a step where previously evaluated points are used to accelerate the search (by minimizing quadratic models) and (ii) addresses the particular cases of bounds on the variables and linear constraints. Our solver is compared to others, and the numerical results confirm its competitiveness in terms of efficiency and robustness.

Keywords: Evolution strategies, constrained optimization, global convergence, extreme barrier function, projection, search step, quadratic models, bound and linear constraints.

1   Introduction

Let us consider a constrained optimization problem of the form

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega \subset \mathbb{R}^n. \end{aligned} \tag{1}$$

In the computational tests reported in this paper we devote particular attention to the case where Ω is defined by a finite number of linear inequalities. Our theory applies to the case where the constraints are nonlinear and some numerical tests for this case are reported as well. In any case, the constraints will be treated as nonrelaxable (meaning that the objective function cannot be evaluated outside the feasible region), and thus the algorithms considered will start feasible and will generate feasible iterates throughout the course of the iterations. The objective function f will be assumed bounded from below in $\mathbb{R}^n$ and Lipschitz continuous near appropriate limit points.

Evolution Strategies (ES's) [46] are evolutionary algorithms designed for global optimization in a continuous space. In [17] we dealt with a large class of ES's, where in each iteration μ points (called parents) are selected as the best in terms of the objective function f of a broader set of λ ($\geq \mu$) points (called offspring), corresponding to the notation (μ, λ)–ES. At the k-th iteration of these ES's, the new offspring $\{y_{k+1}^1, \ldots, y_{k+1}^\lambda\}$ are generated around a weighted mean x_k of the previous parents, corresponding to the notation $(\mu/\mu_W, \lambda)$–ES. The generation process is done by $y_{k+1}^i = x_k + \sigma_k^{\mathrm{ES}} d_k^i$, $i = 1, \ldots, \lambda$, where d_k^i is drawn from a certain distribution and σ_k^{ES} is a chosen step size. One relevant instance of such an ES is CMA-ES [27].

In [17] it has been shown, for unconstrained optimization, how to modify the above mentioned class of ES's to rigorously achieve a form of global convergence, meaning convergence to stationary points independently of the starting point. The modifications in [17] consisted essentially on the reduction of the size of the steps whenever a sufficient decrease condition on the objective function values is not verified. When such a condition is satisfied, the step size can be reset to the step size σ_k^{ES} maintained by the ES's themselves, as long as this latter one is sufficiently large. A number of ways were suggested in [17] to impose sufficient decrease for which global convergence holds under reasonable assumptions. The numerical experiments therein measured the effect of these modifications into CMA-ES [27]. The overall conclusions were that modifying ES's to promote smaller steps when the larger steps are uphill leads to an improvement in the efficiency of the algorithms in the search of a stationary point. Although $(\mu/\mu_W, \lambda)$–ES are non-elitist, our modified versions do introduce some elitism in the sense that the point used to monitor sufficient decrease is the all time best one.

Since the constrained setting poses a number of additional difficulties and technicalities, the paper [17] was confined to unconstrained optimization. In the general context of ES's, various algorithms have been proposed to handle constraints. Coello [11] and Kramer [36] provide a comprehensive survey of the most popular constrained optimization methods currently used within ES's. Most approaches use penalty functions [48], where a term penalizing infeasibility is added to the objective function. Other more sophisticated approaches are based on the use of multiobjective optimization [20] or biologically inspired techniques [21, 47].

In this paper we develop a general globally convergent framework for unrelaxable constraints and make it concrete and operational for the bound and linearly constrained cases. For that purpose, two different approaches are considered. A first one relies on techniques used in directional direct-search methods (see the surveys [13, 33]), where one uses a barrier function to prevent infeasible displacements together with the possible use of directions that conform to the local geometry of the feasible region. The second approach is based first on enforcing all the generated sample points to be feasible, by using a projection mapping of the form:

$$\Phi_\Omega \; : \; \mathbb{R}^n \; \to \; \Omega, \quad \Phi_\Omega^2 \; = \; \Phi_\Omega. \tag{2}$$

The projection is not necessarily the Euclidean one or defined using some other distance, al-

though in the case of bound constraints we will use the ℓ_2-projection (as it is trivial to evaluate) and in the case of linear constraints (meaning that at least one of them is not a simple bound) we will use the ℓ_1-projection (as it reduces to the solution of an LP).

The two approaches above described are computationally compared to some of the best solvers available for minimizing a function without derivatives over bound/linear constraints. All the solvers tested, but a trust-region one, are designed for global optimization in an heuristic sense. For bound-constrained problems, the implementation is enhanced by applying a search step, before the main ES one, based on the minimization of quadratic models built upon previously evaluated points. The numerical results confirm the competitiveness our two approaches in terms of both of efficiency and robustness. For non-linear constraints we did a more preliminary type of testing, essentially just a proof of concept, comparing our best approach to a local direct-search solver (both without any search type step).

The paper is organized as follows. We start by introducing in Section 2 the globally convergent ES's for constrained optimization, explaining how that framework rigorously encompasses what we propose to do in this paper for linearly constrained optimization. Our implementation choices are described in more detail in Section 3. Numerical results for a wide test set of problems are presented in Section 4. At the end we make some concluding remarks in Section 5. By default all norms used in this paper are the ℓ_2 ones.

2 Globally convergent evolution strategies for constrained optimization

The main contribution in [17] is essentially the monitoring of the quality of the sampling procedure by checking if the objective function has been sufficiently decreased. When that is not the case the step size σ_k is reduced and the iteration becomes unsuccessful. Otherwise, the iteration is successful and the step size σ_k might recover the original ES value σ_k^{ES} if this latter one is sufficiently large. There are different ways to impose sufficient decrease conditions in ES's. We will adopt here the version that consists of applying sufficient decrease directly to the weighted mean x_{k+1}^{trial} of the new parents (see (4) below), which has been shown in [17] to yield global convergence for unconstrained optimization without any convexity like assumption and to numerically perform the best among the different versions tested. By sufficient decreasing the objective function at the weighted mean, we mean $f(x_{k+1}^{trial}) \leq f(x_k) - \rho(\sigma_k)$, where $\rho(\cdot)$ is a forcing function [33], i.e., a positive, nondecreasing function satisfying $\rho(\sigma)/\sigma \to 0$ when $\sigma \to 0$.

The extension of the globally convergent ES's to the constrained setting follows a feasible approach, where one starts feasible and then prevent stepping outside the feasible region by means of a barrier approach. The sufficient decrease condition is applied not to f but to the barrier function f_Ω defined by:

$$f_\Omega(x) \;=\; \begin{cases} f(x) & \text{if } x \in \Omega, \\ +\infty & \text{otherwise.} \end{cases} \tag{3}$$

We will follow the terminology introduced in [7] and refer to $f_\Omega(x)$ as the extreme barrier function. Such a function is known as the death penalty function in the terminology of evolutionary algorithms. We consider that ties of $+\infty$ are broken arbitrarily in the ordering of the offspring samples.

3

These globally convergent ES's are described in detail below, in Algorithm 2.1. First we note that the distribution $\mathcal{C}_k$ used to generate the ES directions d_k^i is left unspecified (since, in fact, our global convergence results are not dependant of the choice this of distribution). Despite the extension to constraints, there is a difference from [17] is that the directions used to compute the offspring are not necessarily the randomly generated ES directions, in what can be seen as a modification made in preparation to what comes next. We will denote the directions used to compute the offspring by $\tilde{d}_k^i$. Note that Steps 2 and 3 of Algorithm 2.1 make use of the extreme barrier function (3).

Algorithm 2.1 A class of globally convergent ES's (for unrelaxable constraints)

Initialization: Choose positive integers λ and μ such that $\lambda \geq \mu$. Select an initial $x_0 \in \Omega$ and evaluate $f(x_0)$. Choose initial step lengths $\sigma_0, \sigma_0^{\mathrm{ES}} > 0$ and initial weights $(\omega_0^1, \ldots, \omega_0^\mu) \in S$. Choose constants $\beta_1, \beta_2, d_{\min}, d_{\max}$ such that $0 < \beta_1 \leq \beta_2 < 1$ and $0 < d_{\min} < d_{\max}$. Select a forcing function $\rho(\cdot)$. Set $k = 0$.

Until some stopping criterion is satisfied:

1. **Offspring Generation:** Compute new sample points $Y_{k+1} = \{y_{k+1}^1, \ldots, y_{k+1}^\lambda\}$ such that

$$y_{k+1}^i \;=\; x_k + \sigma_k \tilde{d}_k^i, \; i = 1, \ldots, \lambda, \tag{4}$$

where the directions $\tilde{d}_k^i$'s are computed from the original ES directions d_k^i's (which in turn are drawn from a chosen ES distribution $\mathcal{C}_k$ and scaled if necessary to satisfy $d_{\min} \leq \|d_k^i\| \leq d_{\max}$).

2. **Parent Selection:** Evaluate $f_\Omega(y_{k+1}^i)$, $i = 1, \ldots, \lambda$, and reorder the offspring points in $Y_{k+1} = \{\tilde{y}_{k+1}^1, \ldots, \tilde{y}_{k+1}^\lambda\}$ by increasing order: $f_\Omega(\tilde{y}_{k+1}^1) \leq \cdots \leq f_\Omega(\tilde{y}_{k+1}^\lambda)$.

   Select the new parents as the best μ offspring sample points $\{\tilde{y}_{k+1}^1, \ldots, \tilde{y}_{k+1}^\mu\}$, and compute their weighted mean

$$x_{k+1}^{trial} \;=\; \sum_{i=1}^{\mu} \omega_k^i \tilde{y}_{k+1}^i.$$

   Evaluate $f_\Omega(x_{k+1}^{trial})$.

3. **Imposing Sufficient Decrease:**

   If $f_\Omega(x_{k+1}^{trial}) \leq f(x_k) - \rho(\sigma_k)$, then consider the iteration successful, set $x_{k+1} = x_{k+1}^{trial}$, and $\sigma_{k+1} \geq \sigma_k$ (for example $\sigma_{k+1} = \max\{\sigma_k, \sigma_k^{\mathrm{ES}}\}$).

   Otherwise, consider the iteration unsuccessful, set $x_{k+1} = x_k$ and $\sigma_{k+1} = \bar{\beta}_k \sigma_k$, with $\bar{\beta}_k \in (\beta_1, \beta_2)$.

4. **ES Updates:** Update the ES step length $\sigma_{k+1}^{\mathrm{ES}}$, the distribution $\mathcal{C}_k$, and the weights $(\omega_{k+1}^1, \ldots, \omega_{k+1}^\mu) \in S$. Increment k and return to Step 1.

Due to the sufficient decrease condition, one can guarantee that a subsequence of step sizes will converge to zero. From this property and the fact that the step size is significantly reduced (at least by β_2) in unsuccessful iterations, one proves that there exists a subsequence K of unsuccessful iterates driving the step size to zero (what is referred to as a refining subsequence [6]).

Consequently, assuming boundedness of the sequence of iterates, it is possible to assure the existence of a convergent refining subsequence. We summarize such a result below. The proof can be found in [17], and it is based on the imposition of sufficient decrease by means of a forcing function as in [14, 33].

Lemma 2.1 *Consider a sequence of iterations generated by Algorithm 2.1 without any stopping criterion. Let f be bounded below. There exists a subsequence K of unsuccessful iterates for which $\lim_{k \in K} \sigma_k = 0$. In addition, if the sequence $\{x_k\}$ is bounded, then there exists an x_* and a subsequence K of unsuccessful iterates for which $\lim_{k \in K} \sigma_k = 0$ and $\lim_{k \in K} x_k = x_*$.*

The global convergence is then achieved by establishing that some type of directional derivatives are nonnegative at limit points of refining subsequences along certain limit directions (to be made more precise later). When f is Lipschitz continuous near x_*, one can make use of the Clarke-Jahn generalized derivative along a direction d

$$f^\circ(x_*; d) = \limsup_{\substack{x \to x_*, x \in \Omega \\ t \downarrow 0, x + td \in \Omega}} \frac{f(x + td) - f(x)}{t}.$$

(Such a derivative is essentially the Clarke generalized directional derivative [10], adapted by Jahn [32] to the presence of constraints.) However, for the proper definition of $f^\circ(x_*; d)$, one needs to guarantee that $x + td \in \Omega$ for $x \in \Omega$ arbitrarily close to x_* which is assured if d is hypertangent to Ω at x_*. In the following definition we will use the notation $B(x; \Delta) = \{y \in \mathbb{R}^n : \|y - x\| \le \Delta\}$.

Definition 2.1 *A vector $d \in \mathbb{R}^n$ is said to be a hypertangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point x in Ω if there exists a scalar $\epsilon > 0$ such that*

$$y + tw \in \Omega, \quad \forall y \in \Omega \cap B(x; \epsilon), \quad w \in B(d; \epsilon), \quad and \quad 0 < t < \epsilon.$$

The hypertangent cone to Ω at x, denoted by $T_\Omega^H(x)$, is then the set of all hypertangent vectors to Ω at x. Then, the Clarke tangent cone to Ω at x (denoted by $T_\Omega(x)$) can be defined as the closure of the hypertangent cone $T_\Omega^H(x)$ (when the former is nonempty, an assumption we need to make for global convergence anyway). The Clarke tangent cone generalizes the notion of tangent cone in Nonlinear Programming [45], and the original definition $d \in T_\Omega(x)$ is given below.

Definition 2.2 *A vector $d \in \mathbb{R}^n$ is said to be a Clarke tangent vector to the set $\Omega \subseteq \mathbb{R}^n$ at the point x in the closure of Ω if for every sequence $\{y_k\}$ of elements of Ω that converges to x and for every sequence of positive real numbers $\{t_k\}$ converging to zero, there exists a sequence of vectors $\{w_k\}$ converging to d such that $y_k + t_k w_k \in \Omega$.*

Given a direction v in the tangent cone, possibly not in the hypertangent one, one can consider the Clarke-Jahn generalized derivative to Ω at x_* as the limit

$$f^\circ(x_*; v) = \lim_{d \in T_\Omega^H(x_*), d \to v} f^\circ(x_*; d)$$

(see [7]). A point $x_* \in \Omega$ is considered Clarke stationary if $f^\circ(x_*; d) \ge 0, \forall d \in T_\Omega(x_*)$.

To state the global convergence results it remains to define the notion of refining direction (see [7]), associated with a convergent refining subsequence K, as a limit point of $\{a_k/\|a_k\|\}$ for all $k \in K$ sufficiently large such that $x_k + \sigma_k a_k \in \Omega$, where, in the particular case of taking the weighted mean as the object of evaluation, one has $a_k = \sum_{i=1}^{\mu} \omega_k^i \tilde{d}_k^i$.

2.1 Asymptotic results when derivatives are unknown

In this section we treat constraints as a pure black box in the sense that no information is assumed known about the constrained set Ω, rather than a yes/no answer to the question whether a given point is feasible. The following theorem is in the vein of those first established in [7] for simple decrease and Lipschitz continuous functions (and later generalized in [53] for sufficient decrease and directionally Lipschitz functions).

Theorem 2.1 *Let $x_* \in \Omega$ be the limit point of a convergent subsequence of unsuccessful iterates $\{x_k\}_K$ for which $\lim_{k \in K} \sigma_k = 0$. Assume that f is Lipschitz continuous near x_* with constant $\nu > 0$ and that $T_\Omega^H(x_*) \neq \emptyset$.*

Let $a_k = \sum_{i=1}^{\mu} \omega_k^i \tilde{d}_k^i$. Assume that the directions $\tilde{d}_k^i$'s and the weights ω_k^i's are such that (i) $\sigma_k\|a_k\|$ tends to zero when σ_k does, and (ii) $\rho(\sigma_k)/(\sigma_k\|a_k\|)$ also tends to zero.

If $d \in T_\Omega^H(x_*)$ is a refining direction associated with $\{a_k/\|a_k\|\}_K$, then $f^\circ(x_*; d) \geq 0$.

If the set of refining directions associated with $\{a_k/\|a_k\|\}_K$ is dense in the unit sphere, then x_* is a Clarke stationary point.

Proof. Let d be a limit point of $\{a_k/\|a_k\|\}_K$. Then it must exist a subsequence K' of K such that $a_k/\|a_k\| \to d$ on K'. On the other hand, we have for all k that

$$x_{k+1}^{trial} = \sum_{i=1}^{\mu} \omega_k^i \tilde{y}_{k+1}^i = x_k + \sigma_k \sum_{i=1}^{\mu} \omega_k^i \tilde{d}_k^i = x_k + \sigma_k a_k,$$

and, for $k \in K$ sufficiently large, from the fact that $d \in T_\Omega^H(x_*)$, one has

$$f_\Omega(x_k + \sigma_k a_k) = f(x_k + \sigma_k a_k) > f(x_k) - \rho(\sigma_k).$$

Thus, from the definition of the Clarke generalized derivative,

$$
\begin{aligned}
f^\circ(x_*; d) &= \limsup_{x \to x_*, t \downarrow 0} \frac{f(x + td) - f(x)}{t} \\
&\geq \limsup_{k \in K'} \frac{f(x_k + \sigma_k\|a_k\|(a_k/\|a_k\|)) - f(x_k) + \rho(\sigma_k)}{\sigma_k\|a_k\|} - \frac{\rho(\sigma_k)}{\sigma_k\|a_k\|} - r_k \\
&= \limsup_{k \in K'} \frac{f(x_k + \sigma_k a_k) - f(x_k) + \rho(\sigma_k)}{\sigma_k\|a_k\|} \\
&\geq 0.
\end{aligned}
$$

In the first inequality, we used the fact that $\sigma_k\|a_k\|$ tends to zero on K' (from assumption) and added and subtracted appropriate terms. In the second equality, we used the fact that $\rho(\sigma_k)/(\sigma_k\|a_k\|)$ and r_k both tend to zero on K' (the former from assumption; the latter from the fact that $a_k/\|a_k\| \to d$ on K', see [17]).

To prove the second part, we first conclude from the density of the refining directions on the unit sphere and the continuity of $f^\circ(x_*; \cdot)$ in $T_\Omega^H(x_*)$, that $f^\circ(x_*; d) \geq 0$ for all $d \in T_\Omega^H(x_*)$. Finally, we conclude that $f^\circ(x_*; v) = \lim_{d \in T_\Omega^H(x_*), d \to v} f^\circ(x_*; d) \geq 0$ for all $v \in T_\Omega(x_*)$. ∎

When f is strictly differentiable at x_* (in the sense of Clarke [10], meaning that there exists $\nabla f(x_*)$ such that $f^\circ(x_*; d) = \langle \nabla f(x_*), d \rangle$ for all d), the final result of Theorem 2.1 (as well as of Theorem 2.2 below) reduces to say that the projection of $\nabla f(x_*)$ onto $T_\Omega(x_*)$ is zero.

2.2 Asymptotic results when derivatives are known

Although the approach analyzed in Subsection 2.1 can in principle be applied to any type of constraints, it is obviously more appropriate to the case where one cannot compute the derivatives of the functions algebraically defining the constraints.

Now we consider the case where we can compute tangent cones at points on the boundary of the feasible set Ω. This is the case whenever Ω is defined by $\{x \in \mathbb{R}^n : c_i(x) \leq 0, \ i \in \mathcal{I}\}$ and the derivatives of the functions c_i are known. Two particular cases that appear frequently in practice are bound and linear constraints.

For theoretical purposes, let ϵ be a positive scalar and k_0 a positive integer. Let us also denote by $T_{\Omega, \epsilon, k_0}$ the union of all Clarke tangent cones $T_\Omega(y)$ for all points y at the boundary of Ω such that $\|y - x_k\| \leq \epsilon$ for all $k \geq k_0$:

$$T_{\Omega, \epsilon, k_0} = \cup_{y \in N(\epsilon, k_0)} T_\Omega(y) \quad \text{where} \quad N(\epsilon, k_0) = \{y \in \text{fr}(\Omega) : \|y - x_k\| \leq \epsilon \ \forall k \geq k_0\}.$$

Theorem 2.2 *Let $x_* \in \Omega$ be the limit point of a convergent subsequence of unsuccessful iterates $\{x_k\}_K$ for which $\lim_{k \in K} \sigma_k = 0$. Assume that f is Lipschitz continuous near x_* with constant $\nu > 0$ and that $T_\Omega^H(x_*) \neq \emptyset$.*

Let $a_k = \sum_{i=1}^\mu \omega_k^i \tilde{d}_k^i$. Assume that the directions $\tilde{d}_k^i$'s and the weights ω_k^i's are such that (i) $\sigma_k \|a_k\|$ tends to zero when σ_k does, and (ii) $\rho(\sigma_k)/(\sigma_k \|a_k\|)$ also tends to zero.

If $d \in T_\Omega^H(x_*)$ is a refining direction associated with $\{a_k / \|a_k\|\}_K$, then $f^\circ(x_*; d) \geq 0$.

If the set of refining directions associated with $\{a_k / \|a_k\|\}_K$ is dense in the intersection of $T_{\Omega, \epsilon, k_0}$ with the unit sphere (for some $\epsilon > 0$ and positive integer k_0), then x_* is a Clarke stationary point.

Proof. It has already been shown in Theorem 2.1 that if $d \in T_\Omega^H(x_*)$ is a refining direction associated with $\{a_k / \|a_k\|\}_K$, then $f^\circ(x_*; d) \geq 0$.

The rest of the proof results from the fact that the Clarke tangent cone $T_\Omega(x_*)$ is contained in $T_{\Omega, \epsilon, k_0}$ for any limit point x_* of a subsequence of iterates (and in particular for the subsequence K in the statement of the theorem). Thus, $f^\circ(x_*; v) = \lim_{d \in T_\Omega^H(x_*), d \to v} f^\circ(x_*; d) \geq 0$ for all $v \in T_\Omega(x_*)$. ∎

A known derivative-free technique for handling the constrained case with known constraint gradients is based on computing sets of positive generators for appropriate tangent cones (see [33])[1]. By a set of positive generators of a convex cone, it is meant a set of vectors that spans the cone with nonnegative coefficients. On the other hand, the two known globalization

---

[1]There are a number of globally convergent *hybrid* approaches using penalty or augmented Lagrangian functions (see [39]) or filter techniques (see [8]) *without* attempting to compute positive generators of the appropriated tangent cones related to the nonlinear part of the constraints.

strategies for direct search (for driving the step-size parameter to zero) are generation of iterates on integer/rational lattices and imposition of a sufficient decrease condition for accepting new iterates.

A difficulty when using the first above mentioned strategy for nonlinear constraints is that the positive generators of the tangent cones in consideration can lack of rationality, interfering in the generation of iterates in such lattices. But then, if no tangent cone information is used, the directions have to be densely generated in the whole unit sphere. What makes it possible to derive a result like Theorem 2.2 valid for nonlinear constraints is the use of sufficient decrease (which takes care of the need to drive the step-size parameter to zero). Being free of lattice type requirements, the directions can then be restricted to tangent cones (even if densely generated in such cones, as stated in Theorem 2.2).

An approach based on extreme barrier and the inclusion of positive generators

We also point out that asynchronous parallel directional direct-search methods based on the generating set search framework [33] have been proposed by Griffin, Kolda, and Lewis [24] for linearly-constrained optimization

In the literature of direct-search methods (of directional type) for constraints, one finds approaches specifically developed for the bound or linear constrained cases (see [24, 33, 34, 38]), where positive generators of the appropriated tangent cones are computed and used for what is called polling (i.e., for evaluating the objective function at points of the form $x_k + \sigma_k d$, where d is a positive generator). Although we also address constraints of that type in this paper, we do not want to resort our poll of directions completely to such positive generators as that would not allow to take advantage of the ES random mechanism (Theorem 2.2 would however provide a possible theoretical coverage for such an approach). Instead, we propose to modify the set of directions generated by ES's to include positive generators of appropriate tangent cones. The details will be given in the next section.

The point to make here is that the global convergence result of Theorem 2.1 remains valid as long as the set $\{\tilde{d}_k^i, i = 1, \ldots, \lambda\}$ still verifies Assumptions (i) and (ii). Assumption (i) is trivially satisfied as long as all the positive generators $\tilde{d}_k^i$ are bounded above in norm, which is explicit in the algorithm when $\tilde{d}_k^i = d_k^i$ is an ES randomly generated direction (and can be trivially imposed if $\tilde{d}_k^i$ is a positive generator). The satisfaction of Assumption (ii) is met, for instance, if a_k is bounded below in norm. That in turn depends on the calculation of the all the $\tilde{d}_k^i$'s and on the choice of the weights ω_k^i's, but can always be achieved in the limit case where one weight is set to one and the others to zero.

An approach based on projecting onto the feasible region

The second globally convergent approach is based on projecting onto the feasible domain all the generated sampled points $x_k + \sigma_k d_k^i$, and then taking instead $\Phi_\Omega(x_k + \sigma_k d_k^i)$. We note that projecting onto the feasible region in the context of derivative-free optimization has been already advocated in [41].

This procedure is however equivalent to consider

$$\tilde{d}_k^i = \frac{\Phi_\Omega(x_k + \sigma_k d_k^i) - x_k}{\sigma_k}$$

in the framework of Algorithm 2.1. By substituting all the infeasible generated sampled points by their projections one also conforms the distribution of the offspring to the local geometry of the constraints. When Ω is convex, unlike in the first approach, one does not need here to make use of the extreme barrier function and thus its presence in Steps 2 and 3 of Algorithm 2.1 is innocuous. The details will also be given in the next section.

Again, the global convergence result of Theorem 2.1 remains valid as long as the set $\{\tilde{d}_k^i, i = 1, \ldots, \lambda\}$ still verifies Assumptions (i) and (ii). If we look at Assumption (i), one sees that

$$\sigma_k \|a_k\| = \left\| \sum_{i=1}^{\mu} \omega_k^i [\Phi_\Omega(x_k + \sigma_k d_k^i) - x_k] \right\| \leq \sigma_k \sum_{i=1}^{\mu} \omega_k^i L_{\Phi_\Omega} \|d_k^i\|,$$

since $x_k = \Phi_\Omega(x_k)$, where we assumed that the projection mapping Φ_Ω is Lipschitz continuous with constant $L_{\Phi_\Omega} > 0$. Since the d_k^i's are bounded above in norm, one concludes that $\sigma_k \|a_k\|$ does indeed tend to zero. Note that the projection Φ_Ω is Lipschitz continuous when defined in the best approximation sense using some norm or distance (being the constant L_{Φ_Ω} equal to 1 in the Euclidean/ℓ_2 case). The satisfaction of Assumption (ii) is achieved if a_k is bounded below in norm and similar considerations as in the previous approach apply here too.

3   Implementation choices

In this section, we address linearly constrained problems of the form (1) where Ω is defined as $\{x \in \mathbb{R}^n : Cx \leq d\}$, $C \in \mathbb{R}^{m \times n}$, and $d \in \mathbb{R}^m$, for some positive integer m.

For the approach based on extreme barrier and the inclusion of positive generators

In this approach we form the set of directions $\{\tilde{d}_k^i\}$ by adding positive generators of an appropriated tangent cone to the ES randomly generated directions d_k^i, whenever the current iterate is closer to the boundary of the feasible region.

More specifically, at the current iterate x_k, given $\epsilon_k > 0$, we first identify the ϵ_k-active constraints $I_k = \{i \in \{1, \ldots, m\} : c_i x_k - d_i \geq -\epsilon_k\}$, where c_i denotes the i-th line of C, and then represent by $C_k \in \mathbb{R}^{|I_k| \times n}$ the submatrix of C formed by the rows associated with the ϵ_k-active constraints. The directions to be considered for inclusion are the positive generators D_k of the tangent cone formed at a point where the active constraints are those in I_k. We choose ϵ_k to be $\mathcal{O}(\sigma_k)$ as in [34] (to avoid considering all positive generators for all tangent cones for all $\epsilon \in [0, \epsilon_*]$ where $\epsilon_* > 0$ is independently of the iteration counter as proposed in [38]). We then use the following algorithm from [52] to compute the set D_k of positive generators for corresponding tangent cone (in turn inspired by the work in [38, 3]). Basically, the idea of this algorithm is to dynamically decrease ϵ_k in the search for a set of positive generators of a tangent cone corresponding to a full row rank matrix C_k.

Algorithm 3.1 Calculating the positive generators D_k

Initialization: Choose $\epsilon_k = \min(0.1, 10\sigma_k)$ and $\epsilon_{limit} = \min(0.1, \epsilon_k^2)$.

While $\epsilon > \epsilon_{limit}$

1. Construct the matrix C_k.

2. If $0 < \dim(C_k) < n$ and C_k is full rank, then

   a. Compute a QR factorization of the matrix $C_k^\top$.

   b. Let $Z_k = QR^{-\top}$, $Y_k = I - Z_k C_k$, and stop with $D_k = [\, Z_k \ -Z_k \ Y_k \ -Y_k \,]$.

3. If $\dim(C_k) = 0$, then stop (and return $D_k = [\ ]$), else $\epsilon_k = \epsilon_k/2$.

End While.

The final set of directions $\{\tilde{d}_k^i, i = 1, \ldots, \lambda\}$ is then formed by selecting among $\{d_k^i, i = 1, \ldots, \lambda\} \cup D_k$ those that lead to the best objective function value at the points $x_k + \sigma_k d$ with $d \in \{d_k^i, i = 1, \ldots, \lambda\} \cup D_k$. Our experience tells us that the ES directions are still predominant, representing more than 90% of the overall selected directions. The solver produced in this case will be called ES-LC-B standing for an **E**volution **S**trategy to handle **L**inear **C**onstraints using a **B**arrier approach.

For the approach based on projecting onto the feasible region

In the approach based on projecting onto the feasible region one needs to define the projection mapping Φ_Ω. Given a norm $\|\cdot\|$ and a nonempty closed, convex set Ω, the mapping Φ_Ω can be defined as:

$$\Phi_\Omega(x) \in \operatorname{argmin}\{\|z - x\| : z \in \Omega\}. \tag{5}$$

For purely bound constrained problems, when $\Omega = \{x \in \mathbb{R}^n : l \le x \le u\}$, we will use the ℓ_2-norm since it reduces to a trivial computation. In fact, in the Euclidean case, the projection (5) is unique and simply given by (for $i = 1, \ldots, n$)

$$[\Phi_\Omega(x)]_i = \begin{cases} l_i & \text{if } x_i < l_i, \\ u_i & \text{if } x_i > u_i, \\ x_i & \text{otherwise.} \end{cases}$$

For linearly constrained problems, the Euclidean/ℓ_2 projection (5) reduces to the unique solution of a QP problem with inequality constraints. We will rather use a projection (5) when the norm is the ℓ_1 one as its evaluation requires instead the solution of an LP problem.

Another possibility would be to damp the step and allow the longest displacement along each direction, in other words to compute for each direction $\tilde{d}_k^i$ the largest $\alpha_k^i \in (0, 1]$ such that $y_{k+1}^i = x_k + \alpha_k^i(\sigma_k \tilde{d}_k^i) \in \Omega$. Although such a projection does not require the solution of any auxiliary problem, it depends on the iteration counter and, furthermore, it did not lead to better overall results when compared to the ℓ_1 one.

The related solver will be called ES-LC-P standing for an **E**volution **S**trategy to handle **L**inear **C**onstraints using a **P**rojection approach.

For a search step

Our modified ES algorithms evaluate the objective function at a significantly large number of points at each iteration, independently of its success or unsuccess. In a certain sense, they are even worse than opportunistic direct-search methods were polling is declared successful once a new, better point is found. However, the previously evaluated points can be used in a number of ways to speed up convergence and make ES type algorithms more efficient.

At the beginning of each iteration, a new step called search step can be taken as in the search-poll framework of direct search [9]. For that purpose, a surrogate quadratic model of the objective function f can be minimized in a certain region using previously evaluated points. If the trial point y resulting from this process sufficiently reduces the objective function, meaning if $f_\Omega(y) \leq f(x_k) - \rho(\sigma_k)$, then the search step and the current iteration are declared successful, the trial point is taken ($x_{k+1} = y$), the step size is left unchanged ($\sigma_{k+1} = \sigma_k$), and the ES main iteration step is skipped. If not, the iteration proceeds as in Algorithm 2.1. As in direct-search methods, the search step is optional and has no influence in the global convergence properties since (a) one can still easily prove that there are subsequences of unsuccessful iterations driving the step size to zero (refining subsequences), and (b) the analysis focuses then entirely on subsequences of unsuccessful iterations and those are only attainable by the ES mechanism itself.

To build the surrogate quadratic models, one interpolates over previously evaluated points, following the implementation choices suggested in [15]. The search step is skipped if there are less than $n + 1$ previously evaluated points. If there are less points than needed for complete quadratic interpolation (meaning less than $(n+1)(n+2)/2$), one uses minimum Frobenius norm models [13, Chapter 5]. When there are more than $(n + 1)(n + 2)/2$ previously evaluated, the models are computed by least squares regression, up to a maximum number of points of $(n + 1)(n + 2)$. If there are more points evaluated than $(n + 1)(n + 2)$, 80% of this number are selected as the closest to the current iterate and 20% as the farthest away. The quadratic model is then minimized in a ball (or trust region) $B(x_k; \Delta_k) = \{x \in \mathbb{R}^n : \|x - x_k\|_\infty \leq \Delta_k\}$, centered at x_k with radius $\Delta_k = \theta \sigma_k$ (where θ takes the value 1 if the previous iteration was unsuccessful, or 2 otherwise). The outcome solution of such a minimization is then projected onto the feasible region (using the same projections mentioned above) to yield the trial point y for the search step (and we will have $f_\Omega(y) = f(y)$). The application of such a search step has been shown advantageous for our globally convergent ES's in the unconstrained case [16].

4   Numerical results

We evaluated the performance of the algorithms proposed in Section 2, under the choices described in Section 3, using different solvers, different comparison procedures, and a large collection of bound and linearly constrained problems. Our goal was to assess the efficiency and robustness of the proposed algorithms. At the end of this section we report also preliminary results for non-linear constraints.

4.1   Solvers tested

The solvers used for our numerical comparisons for bound and linear constraints were BCDFO, CMA-ES, MCS, and PSWARM:

- BCDFO [22], Matlab version of Oct. 25, 2011. BCDFO is a local quadratic interpolation-based trust-region algorithm for bound constrained problems.

- CMA-ES (covariance matrix adaptation evolution strategy) for bound constrained optimization, 3.61.beta Matlab version [25, 26]. This constrained version adds to the objective function a penalization term measuring the distance between the current point and its ℓ_2-projection onto the feasible region.

- MCS [30] for bound constrained optimization, 2.0 Matlab version. MCS does a multi-level coordinate search that balances global and local search (the latter using quadratic interpolation).

- PSWARM, the same Matlab version used in [51, 52]. PSWARM implements a polling type direct-search algorithm enhanced by a search step based on swarm optimization for global search. Available for linear constraints.

With the exception of BCDFO, these solvers are designed for global optimization (in an heuristic sense). In the comparative study published in [49], MCS was among the best solvers in terms of both efficiency and robustness. Among the stochastic solvers tested there, CMA-ES and PSWARM have appeared well ranked. BCDFO was developed after this study was carried out but it was shown to perform very well [22].

The default parameters of these four solvers were kept untouched, except the starting point, the initial step size, and the maximal budget, which were chosen the same for all of them including ours.

4.2 Our parameter choices

The parameter choices of Algorithm 2.1 match those of [17] for unconstrained optimization. The values of λ and μ and of the initial weights are the same as in the CMA-ES implementation for unconstrained optimization [25]:

$$
\begin{aligned}
\lambda &= 4 + \text{floor}(3\log(n)), \\
\mu &= \text{floor}(\lambda/2), \\
\omega_0^i &= a_i/(a_1 + \cdots + a_\mu), \text{ with } a_i = \log(\lambda/2 + 1/2) - \log(i), \quad i = 1, \ldots, \mu,
\end{aligned}
$$

where floor($\cdot$) rounds to the nearest integer no larger than the number given. The choices of the distribution $\mathcal{C}_k$ and of the update of σ_k^{ES} are also those used in CMA-ES for unconstrained optimization (see [25]).

The forcing function selected is $\rho(\sigma) = 10^{-4}\sigma^2$. To reduce the step length in unsuccessful iterations we use $\sigma_{k+1} = 0.9\sigma_k$ which corresponds to setting $\beta_1 = \beta_2 = 0.9$. In successful iterations we set $\sigma_{k+1} = \max\{\sigma_k, \sigma_k^{\text{CMA-ES}}\}$ (with $\sigma_k^{\text{CMA-ES}}$ the CMA step size used in ES's). The directions d_k^i, $i = 1, \ldots, \lambda$, were scaled if necessary to obey the safeguards $d_{\min} \leq \|d_k^i\| \leq d_{\max}$, with $d_{\min} = 10^{-10}$ and $d_{\max} = 10^{10}$.

The initial step size is estimated using only the bound constraints, as in [52]: If there is a pair of finite lower and upper bounds for a variable, then σ_0 is set to half of the minimum of such distances, otherwise $\sigma_0 = 20$. The starting point is set to what is suggested in the problem file (or to the origin when there is no suggestion), if such a choice is feasible. When such a choice is not feasible (the majority of the cases), the starting point is the center of the maximum volume ellipsoid inscribed in the feasible region. As in [52], for computing such an ellipsoid we used the software implementation in [54].

In Algorithm ES-LC-P, the projection using the ℓ_1-norm was carried out by solving the corresponding LP problems by the Matlab linprog routine.

The advantage of the search step was clear for the bound constraints case. However, for linear constraints, the search step did not lead to an improvement of the performance and was switched off.

4.3 Test problems (bound and linear constraints)

The problems are coded in AMPL and divided into two groups. The first group includes only pure bound constraints problems and it gathers 114 problems essentially from [4, 31, 40, 43]. These problems have been coded in [51] and all but three are non-convex. The second group includes 107 linearly constrained problems, collected essentially from [1, 50] for the study in [52]. They are not as non-convex as the bound constrained ones since are mostly coming from non-linear programming testing collections. All the solvers were thus interfaced to AMPL. Table 1 shows the distribution of the number of variables and constraints of the problems in the test set. Relatively to the list of test problems made available in [51, 52] we have excluded the bounded constrained problems `lms1a`, `lms1b`, `lms2`, `lms3`, `lms5` due to library linkage and the linearly constrained problems `antenna2`, `powell20` for which none of the solvers were able to find a feasible starting point.

Dimension n of problems	$1 \leq n \leq 10$	$11 \leq n \leq 20$	$21 \leq n \leq 50$	$n \geq 51$
# of problems (bound constraints)	90	10	3	11
# of problems (linear constraints)	72	13	4	18
Number m of constraints	$1 \leq m \leq 10$	$11 \leq m \leq 20$	$21 \leq m \leq 50$	$m \geq 51$
# of problems (linear constraints)	36	17	19	35

Table 1: The distribution of the number of variables and constraints of the problems in the test set.

4.4 Results using performance profiles

Among the existing procedures to compare various solvers over a variety of problems are performance profiles [18]. Given a set of problems $\mathcal{P}$ (of cardinality $|\mathcal{P}|$) and a set of solvers $\mathcal{S}$, the performance profile [18] $\rho_s(\tau)$ of a solver s is defined as the fraction of problems where the performance ratio $r_{p,s}$ is at most τ

$$\rho_s(\tau) = \frac{1}{|\mathcal{P}|}\text{size}\{p \in \mathcal{P} : r_{p,s} \leq \tau\}.$$

The performance ratio $r_{p,s}$ is in turn defined by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in \mathcal{S}\}},$$

where $t_{p,s} > 0$ measures the performance of the solver s when solving problem p (seen as a cost; in this paper the number of function evaluations). The convention $r_{p,s} = +\infty$ is used when the solver s fails to satisfy the convergence test on problem p. Better performance of the solver s, relatively to the other solvers on the set of problems, is indicated by higher values of $\rho_s(\tau)$. In particular, efficiency is measured by $\rho_s(1)$ (the fraction of problems for which solver s performs the best) and robustness is measured by $\rho_s(\tau)$ for τ sufficiently large (the fraction of problems solved by s). Following the suggestion in [18] for a better visualization, we will plot the performance profiles in a $\log_2$-scale (for which $\tau = 1$ will correspond to $\tau = 0$).

It was suggested in [19] to use the same (scale invariant) convergence test for all solvers compared using performance profiles. The convergence test used in our experiments was

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

13

where α is an accuracy level and f_* is an approximation for the optimal value of the problem being tested. We set f_* to the global minimum when it is known, otherwise it was chosen as the best objective function value found by all the tested solvers (including all the runs made for averaging the stochastic solvers) using an extremely large computational budget (a number of function evaluations equal to 500000). Thus, in such a case it makes more sense not to select the accuracy level too small, and our tests were performed with $\alpha = 10^{-2}, 10^{-4}$. The performance profiles were then computed for a maximum of 3000 function evaluations.

The plots in Figures 1–3 depict the performance profiles obtained by the different solvers on the two classes of test problems selected. We begin by noting that for bound-constrained problems (Figures 1–2), one observes a clear advantage in using a search step, and thus we will draw conclusions only based on the right plots (b) of these two figures. One can see that BCDFO performed the best in terms of efficiency and the CMA-ES the worst. In terms of robustness, ES-LC-B and MCS performed the best and BCDFO and CMA-ES the worst. The efficiency of MCS and BCDFO is not surprising since they are based on interpolation models and most of the objective functions tested are smooth. Moreover, both solvers are specifically designed to solve bound-constrained problems. For linearly constrained problems (Figure 3), the projection approach (ES-LC-P) performs, both in efficiency and robustness, better than PSWARM. However, the barrier approach (ES-LC-B) yielded the worst profile.
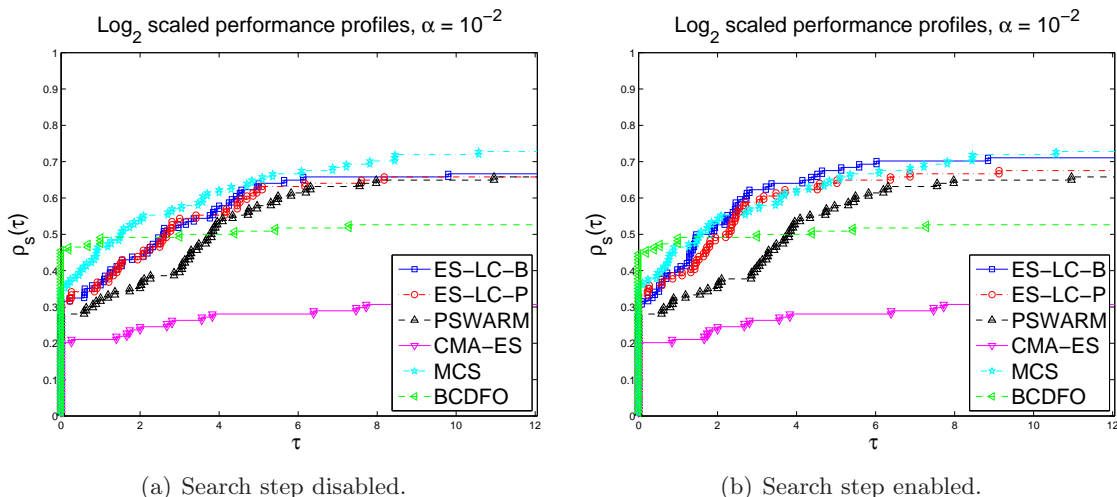

(a) Search step disabled.　　　　　　(b) Search step enabled.

Figure 1: Performance profiles for 114 bound constrained problems using an accuracy level of 10^{-2} (average objective function values for 10 runs).

4.5 Results for data profiles

Data profiles [44] were designed for derivative-free optimization, to show how well a solver performs, given some computational budget, when asked to reach a specific reduction in the objective function value, measured by

$$f(x_0) - f(x) \ \geq \ (1 - \alpha)[f(x_0) - f_L], \tag{6}$$

where $\alpha \in (0, 1)$ is the level of accuracy, x_0 is the initial iterate, and f_L is the best objective value found by all solvers tested for a specific problem within a given maximal computational budget.
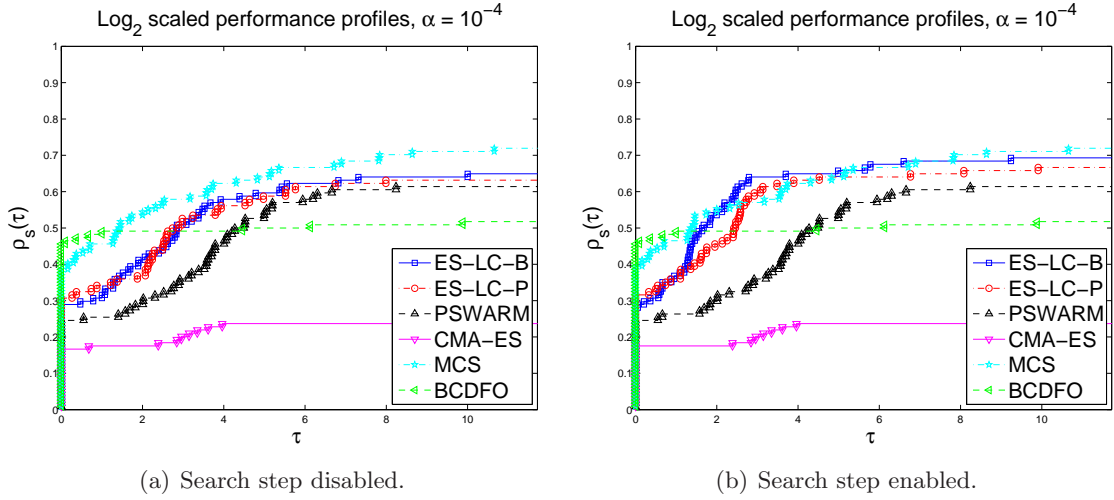
(a) Search step disabled.　　　　　　　(b) Search step enabled.

Figure 2: Performance profiles for 114 bound constrained problems using an accuracy level of 10^{-4} (average objective function values for 10 runs).

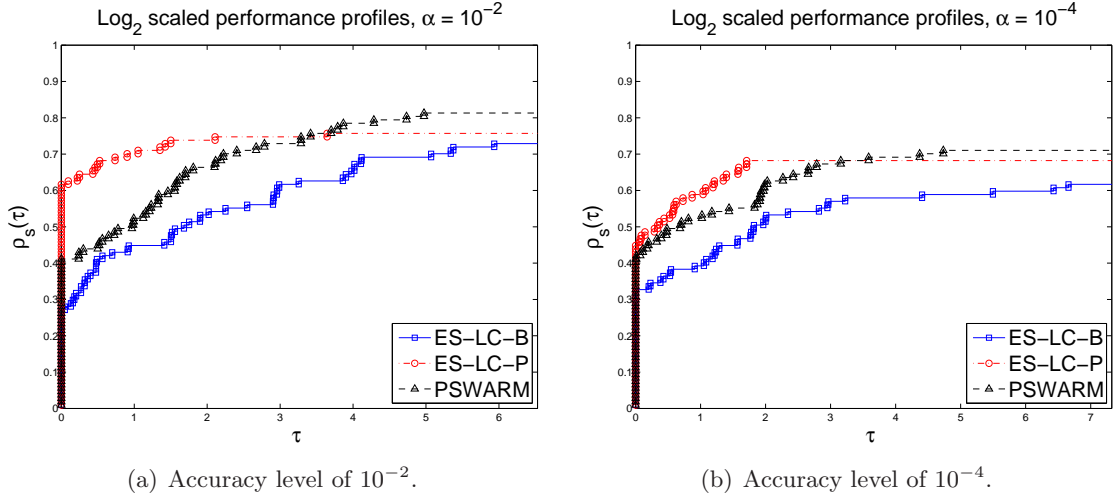(a) Accuracy level of 10^{-2}.　　　　　　(b) Accuracy level of 10^{-4}.

Figure 3: Performance profiles for 107 linearly constrained problems (average objective function values for 10 runs).

In derivative-free optimization, such budgets are typically measured in terms of the number of objective function evaluations. For a stochastic solver, given a number N of runs for the same problem, the value taken for the calculation of f_L is rather set to the average $(\sum_{i=1}^{N} f_L^i)/N$, where f_L^i is the best objective value found during the i-th run of the solver.

Data profiles plot the percentage of problems solved by the solvers under consideration for different values of the computational budget. A data profile is computed, for each solver $s \in \mathcal{S}$, as the percentage of the problems that can be solved within κ function evaluations:

$$d_s(\kappa) \;=\; \frac{1}{|\mathcal{P}|}\text{size}\left\{p \in \mathcal{P} : \frac{t_{p,s}}{n_p + 1} \leq \kappa\right\}, \tag{7}$$

15

where n_p is the number of variables of problem $p \in \mathcal{P}$, and $t_{p,s}$ is the number of function evaluations required by solver $s \in \mathcal{S}$ on problem p to satisfy (6) for a given tolerance α ($t_{p,s} = +\infty$ if the convergence test (6) is not satisfied after the maximum budget of function evaluations). (When the solver is stochastic we plot the average of (7) over the number of runs. These budgets are expressed in number of points ($n_p + 1$) required to form a simplex set, allowing the combination of problems of different dimensions in the same profile. Note that a different function of n_p could be chosen, but $n_p + 1$ is natural in derivative-free optimization, since it is the minimum number of points required to form a positive basis, a simplex gradient, or a model with first-order accuracy.)

We used in our experiments a maximal computational budget consisting of 1500 function evaluations, as we are primarily interested in the behavior of the algorithms for problems where the evaluation of the objective function is expensive. As for the levels of accuracy, we chose two values, $\alpha = 10^{-3}$ and $\alpha = 10^{-7}$. Since the average of the best objective value f_L is chosen as the average best value found by all solvers considered, but under a relatively low maximal computational budget, it makes some sense then to consider a high accuracy level (like 10^{-7} or less).

We now comment on the data profiles obtained by the different solvers on the two classes of problems selected. For bound-constrained problems (Figures 4(b)–5(b)), solvers BCDFO and MCS performed the best for small budgets. For large budgets, ES-LC-B appears as the best solver when looking at both accuracies. Regarding linear constraints (Figure 6), the data profiles show now an advantage for PSWARM which was not observed in the performance profiles where the convergence test was more stringent. As in the performance profiles, ES-LC-P exhibits a better performance than ES-LC-B.

(a) Search step disabled.

(b) Search step enabled.

Figure 4: Data profiles for 114 bound constrained problems using an accuracy level of 10^{-3} (average objective function values for 10 runs).

4.6 Non-linear constraints

We now illustrate the numerical behavior of our globally convergent ES's when the constraints are non-linear. The projection approach is no longer attractive in this case, and thus only the
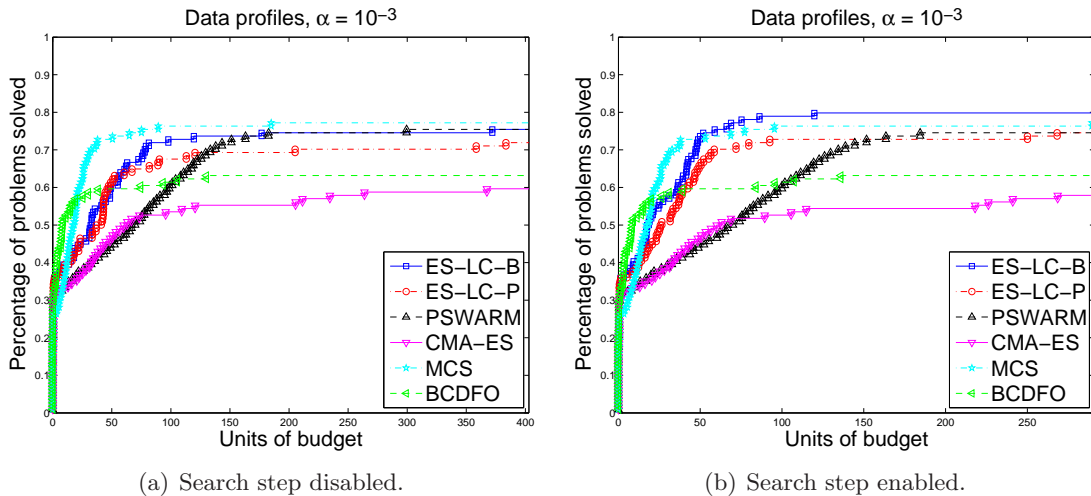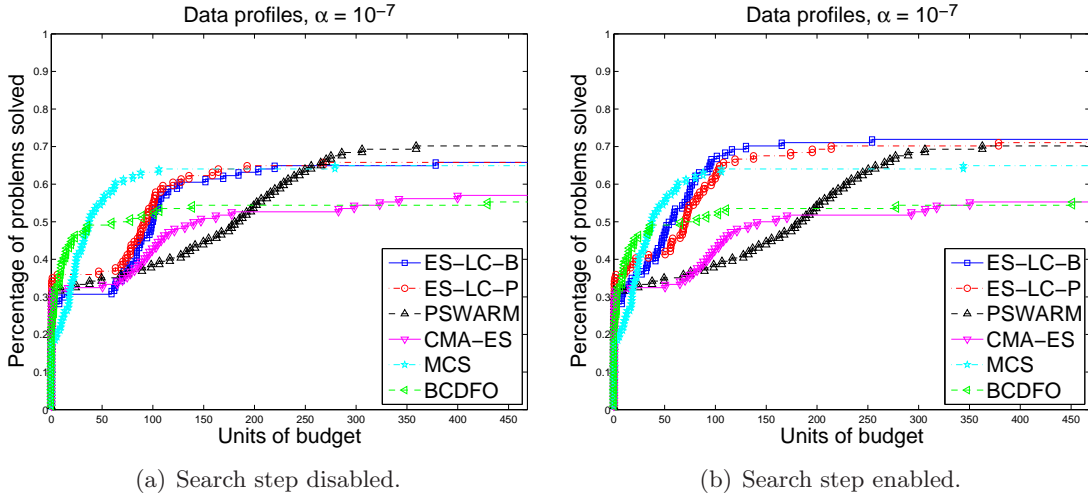
(a) Search step disabled.　　　　　　　　(b) Search step enabled.

Figure 5: Data profiles for 114 bound constrained problems using an accuracy level of 10^{-7} (average objective function values for 10 runs).

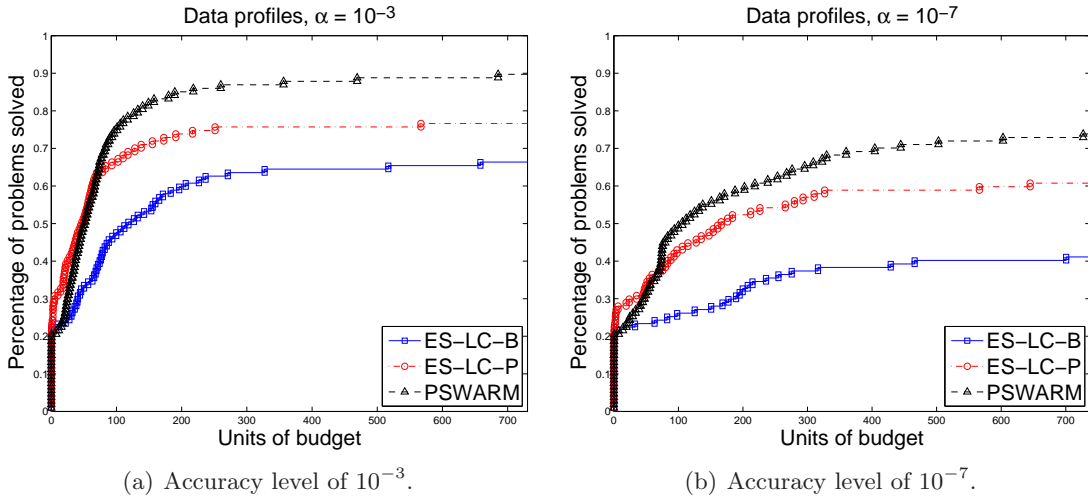(a) Accuracy level of 10^{-3}.　　　　　　(b) Accuracy level of 10^{-7}.

Figure 6: Data profiles for 107 linearly constrained problems (average objective function values for 10 runs).

extreme barrier will be used. The solver tested is then based on Algorithm 2.1 using the extreme barrier and without the generation of positive generators. Simple bounds on the variables are imposed using the ℓ_2-projection. We will call it ES-EB, standing for **E**volution **S**trategy using the **E**xtreme **B**arrier approach.

Our solver will be compared to MADS-EB [7], a direct-search method that uses the extreme barrier function and generates the polling directions asymptotically dense in the unit sphere. We used the MADS-EB implementation given in the NOMAD package [2, 5, 37], version 3.6.1 (a C++ version linked to Matlab via a mex interface), without using any modeling in the search step (option `DISABLE MODELS`).

Since both extreme barrier solvers require a feasible starting point, one applies a common

trick in feasible type methods for Optimization. In fact, it is known that finding a feasible point can be accomplished by computing a global minimizer to the problem of minimizing the constraint violation, in our case defined as

$$
\begin{aligned}
\min \quad & g(x) = \sum_{i \in \mathcal{I}} \max(c_i(x), 0) \\
\text{s.t.} \quad & x \in \mathbb{R}^n.
\end{aligned}
\tag{8}
$$

4.6.1 Test problems

Our test set here is the one used in [28, 29, 35, 42] and comprises 13 well-known test problems G1–G13 (see Table 2). These test problems exhibit a diversity of features and the kind of difficulties that appear in constrained global optimization. In addition to such problems, we added three other engineering optimization problems [28, 12]: PVD the pressure vessel design problem, TCS the tension-compression string problem, and WBD the welded beam design problem. Problems G2, G3, and G8 are maximization problems and were converted to minimization. Problems G3, G5, G11, and WBD contain equality constraints. When a constraint is of the form $c_i^e(x) = 0$, we use the following relaxed inequality constraint instead $c_i(x) = |c_i^e(x)| \leq 10^{-4}$.

Name	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	PVD	TCS	WBD
n	13	20	20	5	4	2	10	2	7	8	2	3	5	4	3	4
m	9	2	1	6	5	2	8	2	4	6	1	1	3	3	4	6

Table 2: Some of the features of the non-linear constrained optimization problems: the dimension n and the number of the constraints m (in addition to the bounds).

4.6.2 Numerical results

Tables 3 and 4 report results (for the average of 10 runs) for both ES-EB and MADS-ES using a maximal budget of 2000 and 20000, respectively. For each problem, we display the best objective value found by each solver. The number of function evaluations is reported also. To be more precise, we display separately the number of constraints violation evaluations $\#g$ used to compute a starting feasible point (when necessary) and then the number of objective function evaluations $\#f$ used from then on. Note that for two problems (G10 and WBD), MADS-EB did not succeed in finding a feasible point for all the ten runs (in which case it used the feasible point found by ES-EB for optimization); see "(F)" in Tables 3 and 4.

For a maximum number of function evaluations of 2000 (Table 3), ES-EB had no difficulty finding a feasible point when the starting point is infeasible, while MADS-ES did not succeed in doing so for problems G10 and WBD. The two solvers were not able to converge to the global optimum value for all the problems, having identified only a stationary point for problems G6, G8, G11, G12, PVD, and TCS. ES-EB converged to a better point compared to MADS-EB for problems G1, G2, G3, G13, and WBD. MADS-EB performed better on problems G4, G5, G6, G7, G9, PVD, and TCS. MADS-EB and ES-EB converged to the same optimum for G6, G8, G11, and G12 with a slight advantage to MADS-EB in terms of the effort spent.

For a larger budget of 20000 function evaluations (Table 4), ES-EB and MADS-EB identified the global optimum for even more problems. The advantage of ES-EB over MADS-EB becomes

Name	Best known	ES-EB			MADS-EB		
	f value	final f	#f	#g	final f	#f	#g
G1	-15	-14.2988	2000	271	-7.82754	2000	583
G2	-0.803619	-0.261765	2000	0	-0.206025	2000	0
G3	-1	-0.0423359	2000	1518	$-6.36481e - 233$	1310	841
G4	-30665.5	-30545.3	2000	12	-30658.3	2000	259
G5	5126.5	5976.79	1653	1531	5609.84	2000	249
G6	-6961.81	-6961.81	1158	56	-6961.81	1863	150
G7	24.3062	36.8157	2000	295	30.3777	2000	603
G8	-0.095825	-0.095825	438	46	-0.095825	343	130
G9	680.63	683.891	2000	0	681.667	2000	0
G10	7049.33	15857	2000	460	7187.31	2000	493 (F)
G11	0.75	0.99998	211	0	0.9998	193	0
G12	-1	-1	247	0	-1	173	0
G13	0.0539498	2.67619	2000	628	2.77742	2000	263
PVD	5868.76	106034	1989	10	6986.89	1552	240
TCS	0.0126653	0.0135891	760	288	0.0127541	711	181
WBD	1.725	2.35943	2000	244	3.03672	1135	232 (F)

Table 3: Results for the extreme barrier approach using a maximal budget of 2000 (average of 10 runs).

clearer using the larger budget. In fact, ES-EB reached a better solution than MADS-EB (if not the global optimum) for 8 problems (more precisely G1, G2, G3, G4, G7, G9, G13, and WBD). Both solvers converged to the same optimum value for four problems (G6, G8, G11, and G12). MADS-EB performed better on the other problems (G5, G10, TCS, and PVD).

5 Conclusions

The goal of this paper was to examine whether globally convergent evolution strategies (ES's) are capable of efficiently handling bounds on the variables and linear constraints. Our motivation originated from the fact that the globally convergent ES's proposed in [17] already yielded encouraging results for unconstrained optimization. We payed particular attention to the need to adapt or conform the generation of the ES offspring to the local geometry of the constraints, and tried to follow the globally convergent principles.

The numerical experiments reported in this paper show that such ES approaches are competitive with state-of-the-art solvers for derivative-free bound and linearly constrained optimization. The results presented for the nonlinear constrained case were not as extensive as those presented for the linear one but have also validated the proposed ES methodology.

In this paper we considered all constraints as unrelaxable, and thus the objective function was never evaluated outside the feasible region. The treatment of relaxable constraints requires the combination of other techniques. In the PhD thesis [16] of the first author, it was analyzed and tested the incorporation of a merit function approach (proposed in [23] for the coverage of relaxable constraints) in our ES methodology, with encouraging results.

Name	Best known f value	ES-EB final f	ES-EB #f	ES-EB #g	MADS-EB final f	MADS-EB #f	MADS-EB #g
G1	-15	-15	15904	271	-7.82761	10093	583
G2	-0.803619	-0.261765	4235	0	-0.206864	11027	0
G3	-1	-0.340501	20000	1518	$-6.36481e-233$	1310	841
G4	-30665.5	-30665.5	6008	12	-30664.9	6666	259
G5	5126.5	5976.79	1653	1531	5609.84	2542	249
G6	-6961.81	-6961.81	1158	56	-6961.81	1863	150
G7	24.3062	24.7203	14606	295	29.9121	7825	603
G8	-0.095825	-0.095825	438	46	-0.095825	343	130
G9	680.63	680.641	11114	0	681.301	3443	0
G10	7049.33	11129.7	20000	460	7186.62	20000	493 (F)
G11	0.75	0.99998	211	0	0.9998	193	0
G12	-1	-1	247	0	-1	173	0
G13	0.0539498	2.52108	5413	628	2.64292	20000	263
PVD	5868.76	35238.9	4063	10	6986.89	1719	240
TCS	0.0126653	0.0135891	760	288	0.0127541	711	181
WBD	1.725	2.35943	2172	244	3.03672	1135	232 (F)

Table 4: Results for the extreme barrier approach using a maximal budget of 20000 (average of 10 runs).

References

[1] GLOBAL Library. http://www.gamsworld.org/global/globallib.htm.

[2] M. A. Abramson, C. Audet, G. Couture, J. E. Dennis Jr., S. Le Digabel, and C. Tribes. The NOMAD project. Software available at http://www.gerad.ca/nomad.

[3] M. A. Abramson, O. A. Brezhneva, J. E. Dennis, and R. L. Pingel. Pattern search in the presence of degenerate linear constraints. *Optim. Methods Softw.*, 23:297–319, 2008.

[4] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *J. Global Optim.*, 31:635–672, 2005.

[5] C. Audet, S. Le Digabel, and C. Tribes. NOMAD user guide. Technical Report G-2009-37, Les cahiers du GERAD, 2009.

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

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

[8] C. Audet and J. E. Dennis Jr. A progressive barrier for derivative-free nonlinear programming. *SIAM J. Optim.*, 20:445–472, 2009.

[9] A. J. Booker, J. E. Dennis Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural and Multidisciplinary Optimization*, 17:1–13, 1998.

[10] F. H. Clarke. *Optimization and Nonsmooth Analysis.* John Wiley & Sons, New York, 1983. Reissued by SIAM, Philadelphia, 1990.

[11] C. A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191:1245–1287, 2002.

[12] C. A. Coello Coello and E. M. Montes. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16:193–203, 2002.

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

[14] I. D. Coope and C. J. Price. Frame based methods for unconstrained optimization. *J. Optim. Theory Appl.*, 107:261–274, 2000.

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

[16] Y. Diouane. *Globally Convergent Evolution Strategies with Application to an Earth Imaging Problem in Geophysics*. PhD thesis, INPT, Toulouse, France, 2014.

[17] Y. Diouane, S. Gratton, and L. N. Vicente. Globally convergente evolution strategies. *Math. Program.*, 2015, to appear.

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

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

[20] C. M. Fonseca and P. J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3:1–16, 1995.

[21] S. Forrest and A. S. Perelson. Genetic algorithms and the immune system. In H.-P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature*, volume 496 of *Lecture Notes in Computer Science*, pages 319–325. Springer, 1991.

[22] S. Gratton, Ph. L. Toint, and A. Tröltzsch. An active-set trust-region method for derivative-free nonlinear bound-constrained optimization. *Optim. Methods Softw.*, 26:873–894, 2011.

[23] S. Gratton and L. N. Vicente. A merit function approach for direct search. *SIAM J. Optim.*, 24:1980–1998, 2014.

[24] J. D. Griffin, T. G. Kolda, and R. M. Lewis. Asynchronous parallel generating set search for linearly-constrained optimization. *SIAM J. Sci. Comput.*, 30:1892–1924, 2008.

[25] N. Hansen. The CMA Evolution Strategy: A Tutorial. June 28, 2011.

[26] N. Hansen, A. S. P. Niederberger, L. Guzzella, and P. Koumoutsakos. A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *IEEE Trans. Evolutionary Computation*, 13:180–197, 2009.

[27] N. Hansen, A. Ostermeier, and A. Gawelczyk. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In L. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms, Pittsburgh*, pages 57–64, 1995.

[28] A. Hedar and M. Fukushima. Derivative-free filter simulated annealing method for constrained continuous global optimization. *J. Global Optim.*, 35:2006, 2004.

[29] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1981.

[30] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *J. Global Optim.*, 14:331–355, 1999.

[31] L. Ingber and B. Rosen. Genetic algorithms and very fast simulated reannealing: A comparison. *Math. Comput. Modelling*, 16:87–100, 1992.

[32] J. Jahn. *Introduction to the Theory of Nonlinear Optimization*. Springer-Verlag, Berlin, 1996.

[33] 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.

[34] T. G. Kolda, R. M. Lewis, and V. Torczon. Stationarity results for generating set search for linearly constrained optimization. *SIAM J. Optim.*, 17:943–968, 2006.

[35] S. Koziel and Z. Michalewicz. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evol. Comput.*, 7:19–44, March 1999.

[36] O. Kramer. A review of constraint-handling techniques for evolution strategies. *Applied Computational Intelligence and Soft Computing*, 2010:1–11, 2010.

[37] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. *ACM Trans. Math. Software*, 37:1–15, 2011.

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

[39] R. M. Lewis and V. Torczon. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM J. Optim.*, 12:1075–1089, 2002.

[40] M. Locatelli. A note on the Griewank test function. *J. Global Optim.*, 25:169–174, 2003.

[41] S. Lucidi, M. Sciandrone, and P. Tseng. Objective-derivative-free methods for constrained optimization. *Math. Program.*, 92:37–59, 2002.

[42] Z. Michalewicz and M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evol. Comput.*, 4:1–32, 1996.

[43] M. Mongeau, H. Karsenty, V. Rouzé, and J.-B. Hiriart-Urruty. Comparison of public-domain software for black box global optimization. *Optim. Methods Softw.*, 13:203–226, 2000.

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

[45] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, Berlin, second edition, 2006.

[46] I. Rechenberg. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.

[47] R. G. Reynolds, Z. Michalewicz, and M. J. Cavaretta. Using cultural algorithms for constraint handling in GENOCOP. In *Evolutionary Programming*, pages 289–305, 1995.

[48] J. T. Richardson, M. R. Palmer, G. E. Liepins, and M. Hilliard. Some guidelines for genetic algorithms with penalty functions. In *Proceedings of the third international conference on Genetic algorithms*, pages 191–197, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[49] L. M. Rios and N. V. Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. *J. Global Optim.*, 56:1247–1293, 2013.

[50] R. J. Vanderbei. Benchmarks for Nonlinear Optimization. `http://www.princeton.edu/~rvdb/bench.html`.

[51] A. I. F. Vaz and L. N. Vicente. A particle swarm pattern search method for bound constrained global optimization. *J. Global Optim.*, 39:197–219, 2007.

[52] A. I. F. Vaz and L. N. Vicente. Pswarm: A hybrid solver for linearly constrained global derivative-free optimization. *Optim. Methods Softw.*, 24:669–685, 2009.

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

[54] Y. Zhang and L. Gao. On numerical solution of the maximum volume ellipsoid problem. *SIAM J. Optim.*, 14:53–76, 2003.