Globally Convergent Evolution Strategies

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

July 9, 2014

Abstract

In this paper we show how to modify a large class of evolution strategies (ES's) for unconstrained optimization to rigorously achieve a form of global convergence, meaning convergence to stationary points independently of the starting point. The type of ES under consideration recombines the parent points by means of a weighted sum, around which the offspring points are computed by random generation. One relevant instance of such an ES is CMA-ES (covariance matrix adaptation ES).

The modifications consist essentially of the reduction of the size of the steps whenever a sufficient decrease condition on the function values is not verified. When such a condition is satisfied, the step size can be reset to the step size maintained by the ES's themselves, as long as this latter one is sufficiently large. We suggest a number of ways of imposing sufficient decrease for which global convergence holds under reasonable assumptions (in particular density of certain limit directions in the unit sphere).

Given a limited budget of function evaluations, our numerical experiments have shown that the modified CMA-ES is capable of further progress in function values. Moreover, we have observed that such an improvement in efficiency comes without weakening significantly the performance of the underlying method in the presence of several local minimizers.

Keywords: Evolution strategy, global convergence, sufficient decrease, covariance matrix adaptation (CMA).

1 Introduction

In recent years, there has been significant and growing interest in derivative-free optimization as represented by talks and publications in mainstream optimization conferences and journals; see [9]. Optimization without derivatives is also the target area of a variety of widely used techniques based on phenomena from nature. Evolutionary algorithms, a popular family of nature-inspired methods, generate iterates through processes modeled on biological evolution, such as

*CERFACS, 42 Avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France (diouane@cerfacs.fr). Support for this author has been provided by Depth Imaging and High Performance Computing TOTAL Exploration & Production, Avenue Larribau, 64018 Pau, France (PI Dr. Henri Calandra).

†ENSEEIHT, INPT, rue Charles Camichel, B.P. 7122 31071, Toulouse Cedex 7, France (serge.gratton@enseeiht.fr).

‡CMUC, Department of Mathematics, University of Coimbra, 3001-501 Coimbra, Portugal (lnv@mat.uc.pt). Support for this author was provided by FCT under grant PTDC/MAT/098214/2008 and by the Réseau Thématique de Recherche Avancée, Fondation de Coopération Sciences et Technologies pour l'Aéronautique et l'Espace, under the grant ADTAO.

recombination, mutation, and selection. Within evolutionary algorithms, evolution strategies (ES's) define search paradigms that specify how each 'generation' leads to the next and seem particularly tailored to optimization with continuous variables; see [6, 17]. Despite addressing the same problem domain, derivative-free optimization and evolution strategies have historically had little connection, in part because of major differences in terminology and notation. The goal of this paper is to help bridge this gap by describing how ideas from derivative-free methods for unconstrained continuous optimization can improve the efficiency and rigorousness of a widely used evolution strategy.

ES's were originally developed in [27] for the unconstrained optimization of a function, $\min_{x\in\mathbb{R}^n} f(x)$, and have been extensively investigated and tested (see, e.g., [6, 17] and the references therein). In the large class of ES's denoted by the notation (μ, λ)–ES, where $\mu > 1$ and λ are integers such that $\mu \leq \lambda$, a certain number λ of points are randomly generated in each iteration, among which μ of them are selected as the best in terms of the objective function f.

Our discussion concentrates on the evolution strategy family denoted in the ES literature by $(\mu/\mu_W, \lambda)$–ES, where the subscript 'W' signals the use of 'recombination' via weights, as described next. Broadly speaking, at iteration k a candidate minimizer x_k is used to produce a generation of λ 'offspring', each consisting of x_k plus a scalar multiple of a random direction; the best μ of these are retained as 'parents' for the next generation ('selection'), and x_{k+1} is taken as a weighted combination ('recombination') of these parents. One relevant instance of such an ES is covariance matrix adaptation ES (CMA-ES) [21].

Derivative-free optimization [9] is a field of nonlinear optimization where methods that do not use derivatives have been developed and rigorously analyzed. There are essentially two classes of algorithms, namely model-based and direct-search methods. However, both are rigorous in the sense of being globally convergent (as it is known in the field of nonlinear optimization).

> By global convergence we mean some form of convergence to some form of first-order stationarity, independent of the starting point. (1)

Model-based and direct-search methods achieve global convergence based on the principle of rejecting steps that are too large and do not provide a certain decrease in the objective function value, retracting the search to a smaller region where the quality of the model or of the sampling eventually allows some progress.

To the best of our knowledge, there are no global convergence results (in the sense of (1)) for ES's, and in particular for $(\mu/\mu_W, \lambda)$–ES; see [5, 15, 22, 23] for asymptotic results about a $(1, \lambda)$–ES, where a single parent is considered and thus 'recombination' is not allowed. The goal of this paper is to add the desirable property of global convergence to ES's by systematically controlling the scalar step size taken in defining each generation of offspring so that a certain decrease is ensured on the objective function.

The technique that we use to globalize such ES's resembles what is done in direct search. In particular, given the type of random sampling used in these ES's, our work is inspired by direct-search methods for nonsmooth functions, where one must use a set of directions asymptotically dense in the unit sphere [4, 29] (more rigorously, a set of directions for which certain limit directions are dense in the unit sphere). Note that random sampling in these ES's would have to be discrete rather than continuous to allow an *integer lattice* underlying structure for the iterates (like in MADS [4]). We will thus use a sufficient decrease condition (as opposed to just a simple decrease) to accept new iterates and ensure global convergence. Our approach is similar to direct-search methods based on *positive spanning sets* for smooth objective functions,

like *frame-based methods* [10] or *generating set searches* [24], or based on asymptotically dense sets for the nonsmooth case [29]. By a sufficient decrease we mean a decrease of the type $f(x_{k+1}) \leq f(x_k) - \rho(\sigma_k)$, where σ_k stands for the step-size parameter and $\rho(\cdot)$ is called a forcing function [24] obeying, in particular $\rho(\sigma)/\sigma \to 0$ when $\sigma \to 0$.

One way of imposing sufficient decrease in the type of ES's under consideration is to apply it directly to the sequence of weighted means. However, ES's are population-based algorithms where a sample set of offspring is generated at every iteration. Other forms of imposing this type of decrease which are also globally convergent involve the maximum value of the best offspring. In fact, requiring a sufficient decrease on the sequence of maximum best offspring values renders a globally convergent algorithm. Furthermore, requiring this maximum value to sufficiently decrease the weighted mean leads also to global convergence.

The paper is organized as follows. We first describe in Section 2 the class of evolution strategies (ES's) to be considered. Then, in Section 3, we show how to modify such algorithms to enable them for global convergence. Section 4 is devoted to the analysis of global convergence of the modified versions of the ES's. Our numerical experiments comparing the different modified versions of CMA-ES [20, 21] are described in Section 5. Finally, in Section 6, we draw some conclusions and describe future work.

2 A Class of Evolution Strategies

As we said in the introduction, we focus on the subclass of (μ, λ)–ES denoted by $(\mu/\mu_W, \lambda)$–ES where, at the k-th iteration, the new offspring $y_{k+1}^1, \ldots, y_{k+1}^\lambda$ are generated around a weighted mean x_k of the previous parents $\tilde{y}_k^1, \ldots, \tilde{y}_k^\mu$. The generation process of the new offspring points is done by $y_{k+1}^i = x_k + \sigma_k^{\text{ES}} d_k^i$, $i = 1, \ldots, \lambda$, where d_k^i is drawn from a certain distribution $\mathcal{C}_k$ and σ_k^{ES} is a chosen step size. The weights used to compute the means belong to the simplex set $S = \{(\omega^1, \ldots, \omega^\mu) \in \mathbb{R}^\mu : \sum_{i=1}^\mu w^i = 1, w^i \geq 0, i = 1, \ldots, \mu\}$, and their values reflect the contribution of each of the previous parents in the weighted mean x_k. The algorithmic description of such a class of ES's is given below.

Algorithm 2.1 A Class of Evolution Strategies

Initialization: Choose positive integers λ and μ such that $\lambda \geq \mu$. Choose an initial x_0, an initial step length $\sigma_0^{\text{ES}} > 0$, an initial distribution $\mathcal{C}_0$, and initial weights $(\omega_0^1, \ldots, \omega_0^\mu) \in S$. 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

$$y_{k+1}^i \;=\; x_k + \sigma_k^{\text{ES}} d_k^i,$$

where d_k^i is drawn from the distribution $\mathcal{C}_k$, $i = 1, \ldots, \lambda$.

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

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

$$x_{k+1} \;=\; \sum_{i=1}^{\mu} \omega^i_k \tilde{y}^i_{k+1}.$$

3. Updates: Update the step length $\sigma^{\rm ES}_{k+1}$, the distribution $\mathcal{C}_{k+1}$, and the weights $(\omega^1_{k+1}, \ldots, \omega^\mu_{k+1}) \in S$. Increment k and return to Step 1.

The specific forms of the updates of the ES step length $\sigma^{\rm ES}_{k+1}$ and of the ES distribution $\mathcal{C}_{k+1}$ are not relevant for the modified globally convergent ES versions to be introduced and analyzed in the next two sections. A particular form of these updates (known as CMA-ES) will be used in the numerical experiments.

3 A Class of Globally Convergent ES's

The main question we address in this paper is how to change Algorithm 2.1, in a minimal way, so that it enjoys some form of convergence properties, while preserving as much as possible the original design and goals. We will target at global convergence in the sense of (1), in other words we would like to prove some limit form of stationarity for any output sequence of iterates generated by the algorithm (i.e., for any realization of the algorithm), and we would like to do this independently of the starting point.

The modifications to the algorithm will be essentially two, and they have been widely used in the field of nonlinear optimization, with and without derivatives. First we need to control the size of the steps taken, and thus we will update separately a step-size parameter σ_k, letting it take the value of $\sigma^{\rm ES}_k$ whenever possible. Controlling the step size is essential as we know that steps used in nonlinear optimization may be too large away from stationarity — an example is Newton's method without a line search, which may take arbitrarily large steps if not started sufficiently close to a problem solution. Secondly we need to impose some form of sufficient decrease on the objective function values to be able to declare an iteration successful and thus avoid a step-size reduction. These two techniques, step-size update and imposition of sufficient decrease on the objective function values, are thus closely related since an iteration is declared unsuccessful and the step size reduced when the sufficient decrease condition is not satisfied. Moreover, this condition involves a function $\rho(\sigma_k)$ of the step size σ_k, where $\rho(\cdot)$ is a forcing function [24], i.e., a positive, nondecreasing function defined in $\mathbb{R}^+$ such that $\rho(t)/t \to 0$ when $t \downarrow 0$ (one can think for instance of $\rho(t) = t^2$).

Since Algorithm 2.1 evaluates the objective function at the offspring sample points but then computes new points around a weighted sum of the parents selected, it is not clear how one can impose sufficient decrease. In fact, there are several ways of proceeding. A first possibility (denoted by mean/mean) is to require the weighted mean to sufficiently decrease the objective function, see the inequality (3) below, which obviously requires an extra function evaluation per iteration.

A second possibility to impose sufficient decrease (referred to as max/max), based entirely on the objective function values already computed for the parent samples, is to require the maximum of these values to be sufficiently decreased, see the inequality (4). Another possibility is to combine these two approaches, asking the new maximum value to reduce sufficiently the value of the previous mean or, vice-versa, requiring the value of the new mean to reduce sufficiently

the previous maximum. The difficulty in proving global convergence in the latter possibility made us consider only the first one, called max/mean, see the inequality (5).

Version mean/mean is clear in the sense that it imposes the sufficient decrease condition directly on the function values computed at the sequence of minimizer candidates, the weighted sums. It is also around these weighted sums that new points are randomly generated. Versions max/max and mean/max, however, operate based or partially based on the function values at the parents samples (on the maximum of those). Thus, in these two versions, one needs to impose a condition of the form (2) below to balance the function values at the parents samples and the function value at the weighted sum. (A certificate of convexity of the objective function would make condition (2) true for any weights in S, but neither such a certificate is realistic when optimizing without derivatives nor would perhaps the type of techniques explored in this paper be the most appropriate under such a scenario.)

The modified form of the ES's of Algorithm 3.1 is described below. Note that one also imposes bounds on all directions d_k^i used by the algorithm. This modification is, however, very mild since the lower bound $d_{\min}$ can be chosen very close to zero and the upper bound $d_{\max}$ set to a very large number. Moreover, one can think of working always with normalized directions which removes the need to impose such bounds.

Algorithm 3.1 A Class of Globally Convergent ES's

Initialization: Choose positive integers λ and μ such that $\lambda \geq \mu$. Select an initial x_0, evaluate $f(x_0)$ in versions mean/mean and max/mean, and set $x_0^\mu = x_0$ for max/max. Choose initial step lengths $\sigma_0, \sigma_0^{\text{ES}} > 0$, an initial distribution $\mathcal{C}_0$, 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

$$y_{k+1}^i \;=\; x_k + \sigma_k d_k^i,$$

where d_k^i is drawn from the distribution $\mathcal{C}_k$ and obeys $d_{\min} \leq \|d_k^i\| \leq d_{\max}$, $i = 1, \ldots, \lambda$.

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

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 $f(x_{k+1}^{trial})$. In versions max/max and max/mean, update the weights, if necessary, such that $(\omega_k^1, \ldots, \omega_k^\mu) \in S$ and

$$f(x_{k+1}^{trial}) \;=\; f\left(\sum_{i=1}^{\mu} \omega_k^i \tilde{y}_{k+1}^i\right) \;\leq\; \sum_{i=1}^{\mu} \omega_k^i f(\tilde{y}_{k+1}^i). \tag{2}$$

3. Imposing Sufficient Decrease:

If (version mean/mean)

$$f(x_{k+1}^{trial}) \leq f(x_k) - \rho(\sigma_k), \tag{3}$$

or (version max/max)

$$f(\tilde{y}_{k+1}^{\mu}) \leq f(x_k^{\mu}) - \rho(\sigma_k), \tag{4}$$

or (version max/mean)

$$f(\tilde{y}_{k+1}^{\mu}) \leq f(x_k) - \rho(\sigma_k), \tag{5}$$

then consider the iteration successful, set $x_{k+1} = x_{k+1}^{trial}$, and $\sigma_{k+1} \geq \sigma_k$ (for example $\sigma_{k+1} = \max\{\sigma_k, \sigma_k^{\mathrm{ES}}\}$).

Set $x_{k+1}^{\mu} = \tilde{y}_{k+1}^{\mu}$ in version max/max.

Otherwise, consider the iteration unsuccessful, set $x_{k+1} = x_k$ (and $x_{k+1}^{\mu} = x_k^{\mu}$ for max/max) 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.

One can see that the imposition of (2) may cost additional function evaluations per iteration. Several iterative schemes (with finite successful termination) can be envisioned to update the weights within Step 2 so that (2) is eventually satisfied. The guarantee of a finite successful termination for such schemes comes from the fact that $\omega_k^1 = 1$, and $\omega_k^i = 0$, $i = 2, \ldots, \mu$, trivially satisfies (2).

The class $(\mu/\mu_W, \lambda)$–ES (Algorithm 2.1) is non-elitist since the next parents are selected only from the current offspring rather than from the combined set of the current parents and offspring all together. The former case would lead to an elitist type of ES since the parents would always be the best points computed so far. However, the modifications that we introduce in Algorithm 3.1 to promote global convergence do lead to some elitism (in the sense that, for instance, in the mean/mean version, we always keep the best mean so far computed).

4 Convergence

Under appropriate assumptions we will now prove global convergence (1) of the modified versions of the considered class of ES's. The objective function f will be assumed bounded from below in $\mathbb{R}^n$ and Lipschitz continuous near appropriate limit points.

As we have seen before, an iteration is considered successful only if it produces a point that has sufficiently decreased some value of f. Insisting on a sufficient decrease will guarantee that a subsequence of step sizes will converge to zero. In fact, since $\rho(\sigma_k)$ is a monotonically nondecreasing function of the step size σ_k, we will see that such a step size cannot be bounded away from zero since otherwise some value of f would tend to $-\infty$. Imposing sufficient decrease will make it harder to have a successful step and therefore will generate more unsuccessful steps. We start thus by showing that there is a subsequence of iterations for which the step-size parameter σ_k tends to zero.

Lemma 4.1 *Consider a sequence of iterations generated by Algorithm 3.1 without any stopping criterion. Let f be bounded below. Then* $\liminf_{k \to +\infty} \sigma_k = 0$.

Proof. Suppose that there exists a $\sigma > 0$ such that $\sigma_k > \sigma$ for all k. If there is an infinite number of successful iterations, this contradicts the fact that f is bounded below.

In fact, since ρ is a nondecreasing, positive function, one has $\rho(\sigma_k) \geq \rho(\sigma) > 0$. Let us consider the three versions separately. In the version mean/mean, we obtain $f(x_{k+1}) \leq f(x_k) - \rho(\sigma)$ for all k, which obviously contradicts the boundedness below of f. In the version max/max, we obtain $f(x^{\mu}_{k+1}) \leq f(x^{\mu}_k) - \rho(\sigma)$ for all k, which also trivially contradicts the boundedness below of f. For the max/mean version, one has

$$f(\tilde{y}^i_{k+1}) \;\leq\; f(x^{\mu}_{k+1}) \;\leq\; f(x_k) - \rho(\sigma_k), \quad i = 1, \ldots, \mu.$$

Thus, multiplying these inequalities by the weights ω^i_k, $i = 1, \ldots, \mu$, and adding them up, lead us to

$$\sum_{i=1}^{\mu} \omega^i_k f(\tilde{y}^i_{k+1}) \;\leq\; f(x_k) - \rho(\sigma_k),$$

and from condition (2) imposed on the weights in Step 2 of Algorithm 3.1, we obtain $f(x_{k+1}) \leq f(x_k) - \rho(\sigma_k)$, and the contradiction is also easily reached.

The proof is thus completed if there is an infinite number of successful iterations. However, if no more successful iterations occur after a certain order, then this also leads to a contradiction. The conclusion is that one must have a subsequence of iterations driving σ_k to zero. ■

From the fact that σ_k is only reduced in unsuccessful iterations and by a factor not approaching zero, one can then conclude the following.

Lemma 4.2 *Consider a sequence of iterations generated by Algorithm 3.1 without any stopping criterion. Let f be bounded below.*

There exists a subsequence K of unsuccessful iterates for which $\lim_{k \in K} \sigma_k = 0$.

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* $\lim_{k \in K} x_k = x_*$.

Proof. From Lemma 4.1, there must exist an infinite subsequence K of unsuccessful iterates for which σ_{k+1} goes to zero. In a such case we have $\sigma_k = (1/\bar{\beta}_k)\sigma_{k+1}$, $\bar{\beta}_k \in (\beta_1, \beta_2)$, and $\beta_1 > 0$, and thus $\sigma_k \to 0$, for $k \in K$, too.

The second part of the lemma is also easily proved by extracting a convergent subsequence of the subsequence K of the first part for which x_k converges to x_*. ■

The above lemma ensures under mild conditions the existence of convergent subsequences of unsuccessful iterations for which the step size tends to zero. Such type of subsequences have been called refining [3]. It has also been suggested in [3] to extract global convergence (1) purely from refining subsequences.

We assume that the function f is Lipschitz continuous near the limit point x_* of a refining subsequence, so that the Clarke generalized derivative [8]

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

exists for all $d \in \mathbb{R}^n$. The point x_* is then Clarke stationary if $f^{\circ}(x_*; d) \geq 0$, $\forall d \in \mathbb{R}^n$.

Global convergence of our various modified ES versions is then proved as suggested in [4], by first establishing that the Clarke derivative is nonnegative along certain limit directions (called

refining directions in [4]) and then by imposing that such directions are dense in the unit sphere. Our overall approach follows what was done in [29] for direct search where such type of analysis was first combined with the use of a forcing function (simplifying considerably the generation of directions which is then free of enforcing sampling points in integer lattices as in [4]).

Our first global convergence result concerns only the mean/mean version.

Theorem 4.1 *Consider the version mean/mean and let* $a_k = \sum_{i=1}^{\mu} \omega_k^i d_k^i$. *Assume that the directions* d_k^i*'s and the weights* ω_k^i*'s are such that* $\|a_k\|$ *is bounded away from zero when* $\sigma_k \to 0$. *Let* x_* *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* $\nu > 0$.

If d *is a limit point of* $\{a_k/\|a_k\|\}_K$, *then* $f^\circ(x_*; d) \geq 0$.

If the set of limit points $\{a_k/\|a_k\|\}_K$ *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

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

and, for $k \in K$,

$$f(x_k + \sigma_k a_k) \;>\; f(x_k) - \rho(\sigma_k).$$

Also, since the directions d_k^i and the weights are bounded above for all k and i, a_k is bounded above for all k, and so $\sigma_k\|a_k\|$ tends to zero when σ_k does.

Thus, from the definition of the Clarke generalized derivative,

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

where, from the Lipschitz continuity of f near x_*,

$$r_k \;=\; \frac{f(x_k + \sigma_k a_k) - f(x_k + \sigma_k\|a_k\|d)}{\sigma_k\|a_k\|} \;\leq\; \nu \left\| \frac{a_k}{\|a_k\|} - d \right\|$$

tends to zero on K'. Finally, since $\|a_k\|$ is bounded away from zero in K',

$$\begin{aligned} f^\circ(x_*; d) &\geq \limsup_{k\in K'} \frac{f(x_k + \sigma_k a_k) - f(x_k) + \rho(\sigma_k)}{\sigma_k\|a_k\|} - \frac{\rho(\sigma_k)}{\sigma_k\|a_k\|} - r_k \\ &= \limsup_{k\in K'} \frac{f(x_k + \sigma_k a_k) - f(x_k) + \rho(\sigma_k)}{\sigma_k\|a_k\|} \\ &\geq 0. \end{aligned}$$

Since the Clarke generalized derivative $f^\circ(x_*; \cdot)$ is continuous in its second argument [8], it is then evident that if the set of limit points $\{a_k/\|a_k\|\}_K$ is dense in the unit sphere, $f^\circ(x_*; d) \geq 0$ for all $d \in \mathbb{R}^n$. ■

When f is strictly differentiable at x_* (in the sense of Clarke [8], meaning that there exists $\nabla f(x_*)$ such that $f^\circ(x_*;d) = \langle \nabla f(x_*), d\rangle$ for all d) we immediately conclude that $\nabla f(x_*) = 0$.

Let us now discuss the assumptions of Theorem 4.1. First, we should point out that the assumption regarding the directions a_k, in particular their density in the unit sphere, applies to a given refining subsequence K and not to the whole sequence of iterates. However, such a strengthening of the requirements on the density of the directions seems necessary for these type of directional methods (see [4, 29]).

Then, the question that arises concerns the density in general of the a_k's in the unit sphere. For the purpose of this discussion, and to keep things simple, let us assume that the weights are fixed for all k (which is a valid choice for Theorem 4.1). Let us assume also that d_k^i's are drawn from a multivariate normal distribution with mean 0 and covariance matrix C. The direction $a_k = \sum_{i=1}^{\mu} \omega^i d_k^i$ is then a realization of a random vector A following a multivariate normal distribution with mean 0 and covariance matrix $\sum_{i=1}^{\mu} (\omega^i)^2 C$. Then, for any $y \in \mathbb{R}^n$ such that $\|y\| = 1$ and for any $\delta \in (0,1)$, there exists a positive constant η such that

$$P(\cos(A/\|A\|, y) \geq 1 - \delta) \ \geq \ \eta$$

since the calculation of this probability results from integrating the Gaussian density function over a set $\{y \in \mathbb{R}^n : \cos(A/\|A\|, y) \geq 1 - \delta\}$ of nonzero Lebesgue measure. This property guarantees us the density of the a_k's in the unit sphere (with probability one).

Finally, under the random generation framework of the previous paragraph one can also see that we could fix an $\epsilon > 0$ (preferably small) at the initialization of the algorithm and then re-sample the d_k^i's again whenever $\|a_k\| < \epsilon$. The density of the a_k's in the unit sphere (with probability one) would then result from the fact that, for the same reasons, for any $y \in \mathbb{R}^n$ such that $\|y\| = 1$ and for any $\delta \in (0,1)$, there would still exist a positive constant η such that

$$P(\cos(A/\|A\|, y) \geq 1 - \delta, \|A\| \geq \epsilon) \ \geq \ \eta.$$

Now, we prove global convergence for the two other versions (max/max and max/mean).

Theorem 4.2 *Consider the versions max/max and max/mean. Let x_* 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 $\nu > 0$.*

If d is a limit point of $\{d_k^{i_k}/\|d_k^{i_k}\|\}_K$, where $i_k \in \operatorname{argmax}_{1\leq i\leq \mu} f(y_{k+1}^i)$, then $f^\circ(x_*;d) \geq 0$.

If, for each $i \in \{1,\ldots,\mu\}$, the set of limit points $\{d_k^i/\|d_k^i\|\}_K$ is dense in the unit sphere, then x_* is a Clarke stationary point.

Proof. The proof follows along the same lines as the proof of the mean/mean version. In the max/max case, one has for $k \in K$,

$$f(\tilde{y}_{k+1}^\mu) \ > \ f(x_k^\mu) - \rho(\sigma_k),$$

which implies for a certain i_k that

$$f(y_{k+1}^{i_k}) \ = \ f(\tilde{y}_{k+1}^\mu) \ > \ f(x_k^\mu) - \rho(\sigma_k).$$

Now, notice that $x_{k+1}^\mu = x_k^\mu = \cdots = x_{k-p_k}^\mu$, where $k - p_k - 1$ is the index of the last successful iteration before k. Thus,

$$f(y_{k+1}^{i_k}) > \ f(x_{k-p_k}^\mu) - \rho(\sigma_k) \ \geq \ f(\tilde{y}_{k-p_k}^i) - \rho(\sigma_k), \quad i = 1,\ldots,\mu.$$

Multiplying these inequalities by the weights $\omega^i_{k-p_k-1}$, $i = 1, \ldots, \mu$, and adding them up implies

$$f(y^{i_k}_{k+1}) \; > \; \sum_{i=1}^{\mu} \omega^i_{k-p_k-1} f(\tilde{y}^i_{k-p_k}) - \rho(\sigma_k),$$

Condition (2) imposed on the weights in Step 2 of Algorithm 3.1 with k replaced by $k - p_k - 1$ implies

$$f(y^{i_k}_{k+1}) \; > \; f\left(\sum_{i=1}^{\mu} \omega^i_{k-p_k-1} \tilde{y}^i_{k-p_k}\right) - \rho(\sigma_k).$$

Since $\sum_{i=1}^{\mu} \omega^i_{k-p_k-1} \tilde{y}^i_{k-p_k} = x^{trial}_{k-p_k} = x_{k-p_k} = x_k$ (because $k-p_k-1$ is successful and $k-p_k, \ldots, k$ are unsuccessful) and $y^{i_k}_{k+1} = x_k + \sigma_k d^{i_k}_k$, we arrive at

$$f(x_k + \sigma_k d^{i_k}_k) \; > \; f(x_k) - \rho(\sigma_k). \tag{6}$$

(If there is no successful iteration before the k-th one, then, since $x_0 = x^{\mu}_0$, we will directly obtain (6).)

Note that in the max/mean version we arrive directly at $f(x_k + \sigma_k d^{i_k}_k) > f(x_k) - \rho(\sigma_k)$.

From this point, and for both cases (max/max and max/mean), the proof is nearly identical to the proof of Theorem 4.1 (in particular note that $d^{i_k}_k$ is forced to be bounded away from zero by Algorithm 3.1). ■

Again, when f is strictly differentiable at x_*, we conclude that $\nabla f(x_*) = 0$. In Theorem 4.2 one also has the same issue regarding the density of the directions on the unit sphere being assumed for a given refining subsequence K rather then for the whole sequence of iterates.

5 Numerical Results

We conduct a number of numerical experiments to measure the effect of our modifications into ES's. We are mainly interested in observing the changes that occur in ES's in terms of an efficient and robust search of stationarity. We choose CMA-ES [20, 21] as our evolutionary strategy, on top of which we test our globally convergent modifications. CMA-ES appeared well ranked in a comparative study published in [28], among the tested stochastic solvers. In addition, we then add a comparison with a direct-search method (MADS, mesh adaptive direct search [4]).

5.1 CMA-ES

In CMA-ES [20] the distributions $\mathcal{C}_k$ are multivariate normal distributions. The weights are kept constant and besides belonging to the simplex S they also satisfy $\omega^1 \geq \cdots \geq \omega^{\mu} > 0$. Briefly speaking and using the notation of our paper, CMA-ES updates the covariance matrix of $\mathcal{C}_k$ as follows:

$$C^{\text{CMA-ES}}_{k+1} \;\; = \;\; (1 - c_1 - c_\mu)\, C^{\text{CMA-ES}}_k + c_1 (p^c_{k+1})(p^c_{k+1})^\top + c_\mu \sum_{i=1}^{\mu} \omega_i \, (d^i_k)(d^i_k)^\top,$$

where c_1, c_μ are positive constants depending on n, and $p^c_{k+1} \in \mathbb{R}^n$ is the current state of the so-called evolution path, updated iteratively as

$$p^c_{k+1} \; = \; (1 - c_C) p^c_k + h_\sigma [c_C (2 - c_C) \mu_f]^{\frac{1}{2}} (x_{k+1} - x_k) / \sigma^{\text{CMA-ES}}_k,$$

where $p_0^c = 0$ and c_C is a positive constant depending on n (see [16] for the choice of the scaling factor h_σ). One can see that $C_{k+1}^{\text{CMA-ES}}$ is updated by a sum of rank-one matrices. One of the rank-one terms involves the mean difference (from current to past iteration) and the remaining ones the directions generated at the current iteration. The step length of CMA-ES is defined as follows:

$$\sigma_{k+1}^{\text{CMA-ES}} \;=\; \sigma_k^{\text{CMA-ES}} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_{k+1}^\sigma\|}{E\|\mathcal{N}(0,I)\|} - 1\right)\right),$$

where $E\|\mathcal{N}(0,I)\| = \sqrt{2}\Gamma(\frac{n+1}{2})/\Gamma(\frac{n}{2})$ is the expectation of the ℓ_2 norm of an $N(0,I)$ distributed random vector, c_σ, d_σ are positive constants, and $p_{k+1}^\sigma \in \mathbb{R}^n$ is the current state of the so-called conjugate evolution path (see [20]).

5.2 Algorithmic Choices for the Modified CMA-ES Versions

In this subsection, we list the parameters and updates chosen in Algorithm 3.1. The values of λ and μ and of the initial weights are the same as in the CMA-ES implementation in [16]:

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

where floor$(\cdot)$ rounds to the nearest integer no larger than the number given. The values for c_1, c_μ, c_C, c_σ, and d_σ are chosen also as in the CMA-ES implementation (see [16]) as

$$\begin{aligned}
c_1 &= 2/((n+1.3)^2 + \mu_f),\\
c_\mu &= \min\{1 - c_1, 2(\mu_f - 2 + 1/\mu_f)/((n+2)^2 + \mu_f)\},\\
c_C &= (4 + \mu_f/n)/(n + 4 + 2\mu_f/n),\\
c_\sigma &= (\mu_f + 2)/(n + \mu_f + 5),\\
d_\sigma &= 1 + 2\max\{0, [(\mu_f - 1)/(n+1)]^{\frac{1}{2}} - 1\} + c_\sigma, \text{ with}\\
\mu_f &= (\omega_0^1 + \cdots + \omega_0^\mu)^2/((\omega_0^1)^2 + \cdots + (\omega_0^\mu)^2).
\end{aligned}$$

The initial step length parameters are set to $\sigma_0 = \sigma_0^{\text{CMA-ES}} = 1$. The forcing function selected is $\rho(\sigma) = 10^{-4}\sigma^2$.

To reduce the step length in unsuccessful iterations we use $\sigma_{k+1} = 0.5\sigma_k$ which corresponds to setting $\beta_1 = \beta_2 = 0.5$. In successful iterations, we use $\sigma_{k+1} = \max\{\sigma_k, \sigma_k^{\text{CMA-ES}}\}$, in an attempt to reset the step length to the ES one whenever possible.

The directions d_k^i, $i = 1, \ldots, \lambda$, are drawn from the multivariate normal distribution $\mathcal{C}_k$ updated by CMA-ES, scaled if necessary to obey the safeguards $d_{\min} \le \|d_k^i\| \le d_{\max}$, with $d_{\min} = 10^{-10}$, $d_{\max} = 10^{10}$. In our experiments, we have never seen a run where there was a need to impose these safeguards.

Updating the weights in Step 2 of Algorithm 3.1 to enforce (2) was not included in the runs reported in this paper. On the one hand, we wanted the least amount of changes in CMA-ES. On the other hand, such an update of the weights in Step 2 did not seem to have a real impact on the results for versions max/max and mean/max, perhaps due to the convexity near the solutions present in many of the problems.

5.3 Test Set

Our test set $\mathcal{P}$ is the one suggested in [26] and comprises 22 nonlinear vector functions from the CUTEr collection [14]. The problems in $\mathcal{P}$ are defined by a vector (k_p, n_p, m_p, s_p) of integers. The integer k_p is a reference number for the underlying CUTEr vector function, n_p is the number of variables, m_p is the number of components $F_1, \ldots, F_{m_p}$ of the corresponding vector function F. The integer $s_p \in \{0, 1\}$ defines the starting point via $x_0 = 10^{s_p} x_s$, where x_s is the standard CUTEr starting point for the corresponding function. According to [26], the use of $s_p = 1$ is helpful for testing solvers from a more remote starting point since the standard starting point tends to be close to a solution for many of the problems. The test set $\mathcal{P}$ is formed by 53 different problems for which n_p and m_p satisfy

$$2 \leq n_p \leq 12, \quad 2 \leq m_p \leq 65, \quad p = 1, \ldots, 53.$$

Table 1 contains the distribution of n_p across the problems. For other details see [26].

n_p	2	3	4	5	6	7	8	9	10	11	12
Number of problems	5	6	5	4	4	5	6	5	4	4	5

Table 1: The distribution of n_p in the test set.

The test problems have been considered in four different types, each having 53 instances: smooth (least squares problems obtained from applying the ℓ_2 norm to the vector functions); nonstochastic noisy (obtained by adding oscillatory noise to the smooth ones); piecewise smooth (as in the smooth case but using the ℓ_1 norm instead); stochastic noisy (obtained by adding random noise to the smooth ones).

5.4 Results using Data and Performance Profiles

To compare our modified CMA-ES versions among each other and against the pure one, we chose to work with two types of profiles, data and performance profiles.

5.4.1 Data Profiles

Data profiles [26] were designed for derivative-free optimization and show how well a solver performs, given some computational budget, when asked to reach a specific reduction in the objective function value, measured by

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

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. In derivative-free optimization, such budgets are typically measured in terms of the number of objective function evaluations.

Data profiles plot the percentage of problems solved by the solvers under consideration for different values of the computational budget. These budgets are expressed in terms of the number of points $(n + 1)$ 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 could be chosen,

but $n+1$ 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).

In our experiments we use a maximal computational budget of $50n$ 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 best objective value f_L is chosen as the best value found by all solvers considered, but under a relatively low maximal computational budget, it makes some sense to consider a high accuracy level (like 10^{-7} or less).

5.4.2 Performance Profiles

Given a set of problems $\mathcal{P}$ (of cardinality $|\mathcal{P}|$) and a set of solvers $\mathcal{S}$, the performance profile [12] $\rho_s(\tau)$ 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 here as a cost, like number of function evaluations). 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 [12] 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 [13] to use the same (scale invariant) convergence test for all solvers compared using performance profiles. The convergence test used in our experiments is

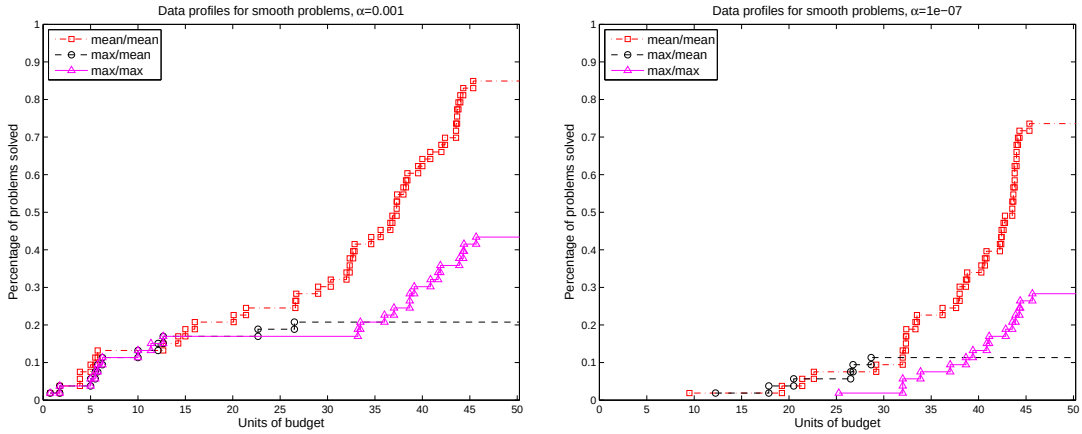
$$f(x) - f_* \ \leq \ \alpha(|f_*| + 1),$$

where α is an accuracy level and f_* is an approximation for the optimal value of the problem being tested. The convention $r_{p,s} = +\infty$ is used when the solver s fails to satisfy the convergence test on problem p. We compute f_* as the best objective function value found by the four CMA-ES solvers (our three modified versions and the pure one) using an extremely large computational budget (a number of function evaluations equal to 500000). Thus, in this case, and as opposed to the data profiles 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 1500 function evaluations.

5.4.3 The Results

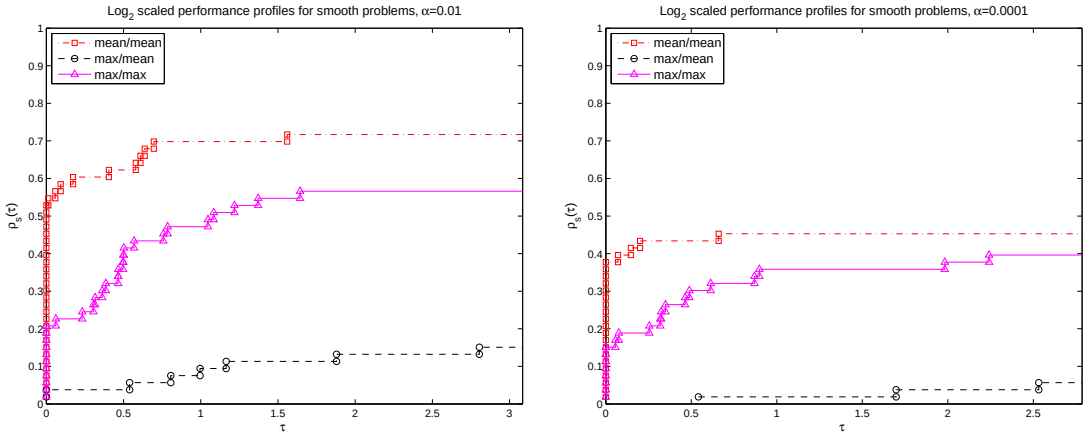
First we have compared the three modified versions of CMA-ES (mean/mean, max/max, and max/mean) among each other. Our experiments have shown that the mean/mean version emerges as the best one. We report here only the results for the class of smooth problems

of Section 5.3 (see Figure 1 for the corresponding two data profiles and Figure 2 for the corresponding two performance profiles), since the results for the other three classes of problems of Section 5.3 follow a similar trend. Note that from the performance profiles of Figure 2 it also becomes clear that the version max/mean performs poorly, an effect we have observed in particular for unimodal functions.



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

Figure 1: Data profiles computed for the set of smooth problems, considering the two levels of accuracy, 10^{-3} and 10^{-7} (for the three modified versions).

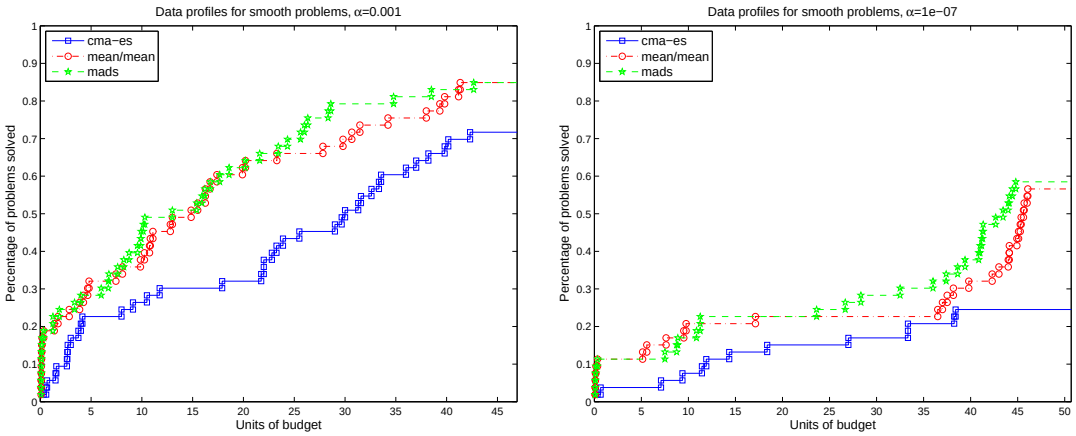


(a) Accuracy level of 10^{-2}. (b) Accuracy level of 10^{-4}.

Figure 2: Performance profiles computed for the set of smooth problems with a logarithmic scale, considering the two levels of accuracy, 10^{-2} and 10^{-4} (for the three modified versions).

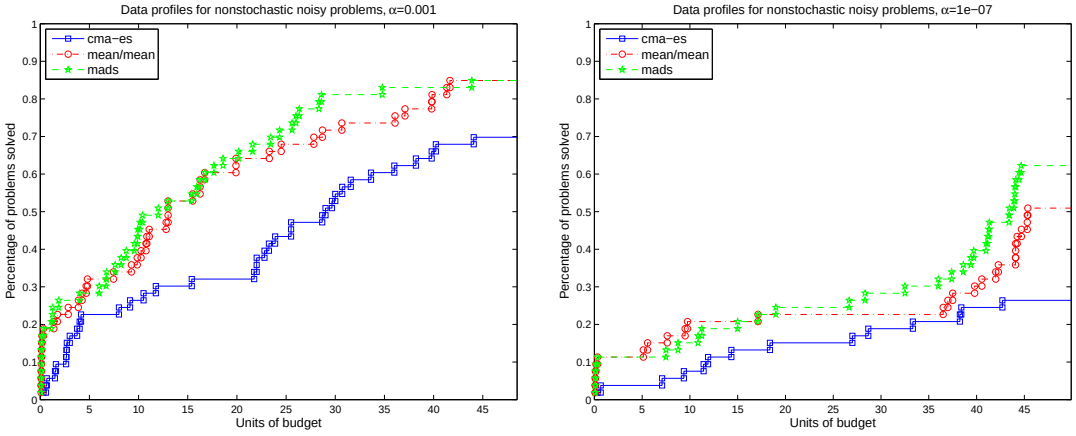
Next, we compare the pure and the mean/mean CMA-ES versions with MADS (mesh adaptive direct search) for which we use the implementation given in the NOMAD package [1, 2, 25], version 3.6.1 (C++ version linked to Matlab via a mex interface), where we enable the option `DISABLE MODELS`, meaning that no modeling is used in MADS, both in the search step and in the construction of directions (and their order of usage) in the poll step.

Figures 3–6 report the data profiles obtained by the mean/mean and pure versions and by MADS, for the four types of problems, considering the two different levels of accuracy, $\alpha = 10^{-3}$ and $\alpha = 10^{-7}$ (Figure 3: smooth problems; Figure 4: nonstochastic noisy problems; Figure 5: piecewise smooth problems; Figure 6: stochastic noisy problems).



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

Figure 3: Data profiles computed for the set of smooth problems, considering the two levels of accuracy, 10^{-3} and 10^{-7}.

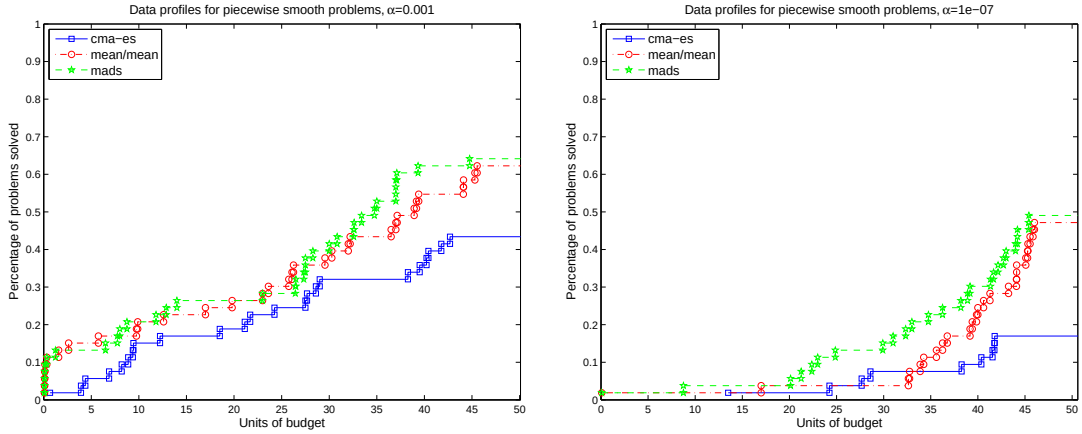


(a) Accuracy level of 10^{-3}. (b) Accuracy level of 10^{-7}.

Figure 4: Data profiles computed for the set of nonstochastic noisy problems, considering the two levels of accuracy, 10^{-3} and 10^{-7}.

Figures 7–10 report performance profiles obtained by the mean/mean and pure versions and by MADS, for the four types of problems, considering the two different levels of accuracy, $\alpha = 10^{-2}$ and $\alpha = 10^{-4}$ (Figure 7: smooth problems; Figure 8: nonstochastic noisy problems; Figure 9: piecewise smooth problems; Figure 10: stochastic noisy problems).

MADS exhibits a slightly better performance than the mean/mean version in the data profiles (which test smaller budgets). However, when the budget is larger, as it is the case in the



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

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

Figure 5: Data profiles computed for the set of piecewise smooth problems, considering the two levels of accuracy, 10^{-3} and 10^{-7}.

performance profiles, the mean/mean version performs roughly the same as MADS in efficiency but better in robustness. The advantage of the mean/mean version over the pure one is clear, either in the data or in the performance profiles, with the exception of the piecewise problems where the pure version overcomes in terms of robustness both the mean/mean version and MADS (see the corresponding performance profiles).

5.5 Some Global Optimization Experiments

In this section we assess the impact of our modifications on the ability of CMA-ES to identify the global minimum on problems with a high number of different local minimizers.

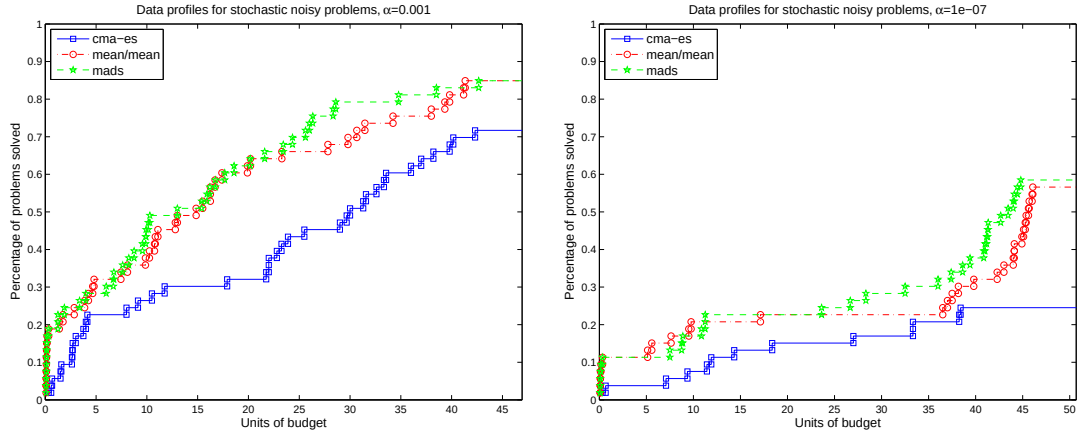
We recall that the mean/mean version exhibited the best performance among the three modified versions of CMA-ES on the test set mentioned in Section 5.3. Therefore, in this section we will report a comparison of CMA-ES only against this version.

The test set is composed of the 19 highly multi-modal problems used in [19, 18], where the last 9 are noisy (see Tables 2–3). We select dimensions $n = 10, 20$, and, for each dimension, population sizes of $\lambda = 2n, 10n$. For each case and using a large maximal computational budget, we run our mean/mean CMA-ES version and pure CMA-ES, from 20 different starting points randomly chosen using the Matlab function `randn`. We then compute the median of all the 20 'optimal' values found for each algorithm as well as the median of the respective number of function evaluations taken.

Problem Number	1	2	3	4	5	6	7	8	9	10
Problem index in [19]	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}

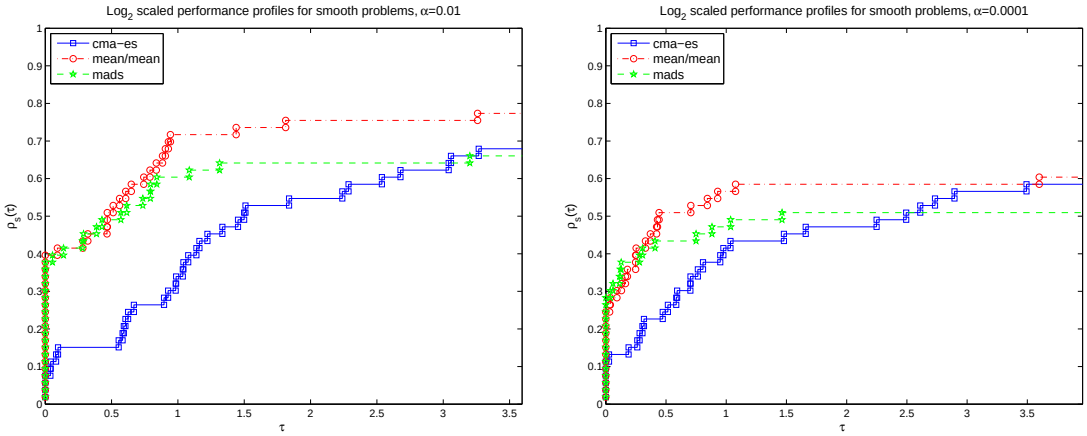
Table 2: Noiseless problems.

Each run terminates when the function value falls below a certain fitness value, chosen as $f_* + 10^{-7}$, where f_* is the optimal value of the corresponding problem, or when the number of function evaluations reaches 250000. To avoid division by large numbers we also stop a run



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

Figure 6: Data profiles computed for the set of stochastic noisy problems, considering the two levels of accuracy, 10^{-3} and 10^{-7}.

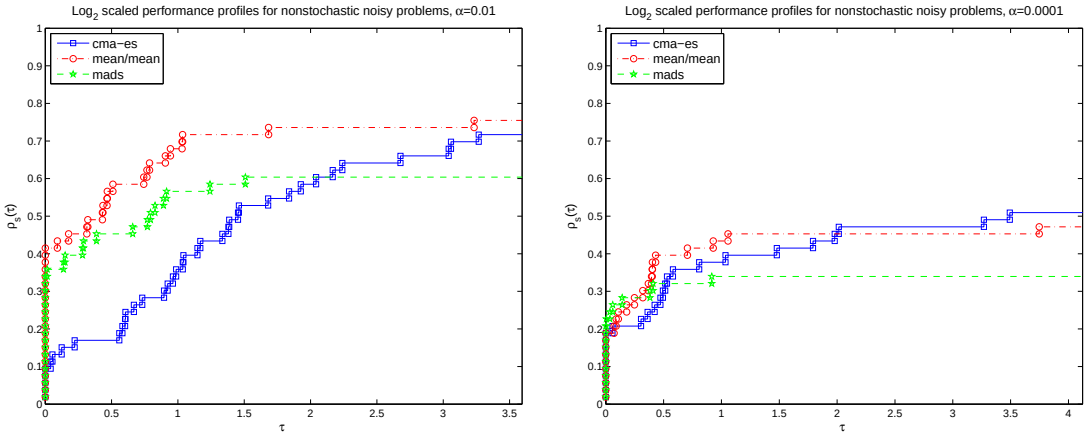


(a) Accuracy level of 10^{-2}. (b) Accuracy level of 10^{-4}.

Figure 7: Performance profiles computed for the set of smooth problems with a logarithmic scale, considering the two levels of accuracy, 10^{-2} and 10^{-4}.

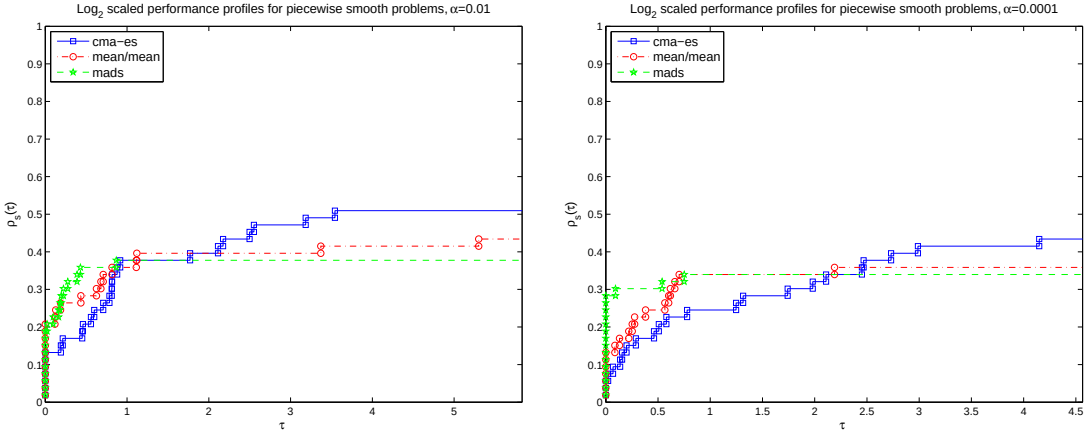
once σ_k becomes smaller than 10^{-10}. It must be made clear that this last criterion makes our versions (in particular the mean/mean one) more parsimonious in terms of function evaluations but it may also possibly restrict the search of the global minimum. Note also that the budget is therefore large and the tolerances small since we are interested in observing the asymptotic ability to determine a global minimum (such choices are not likely to be affordable in practical application problems where the objective function is expensive to evaluate).

Figures 11(a), 12(a), 13(a), and 14(a) show the median best objective value obtained by the mean/mean and the pure CMA-ES versions, as well as the global optimal value, for all problem dimensions and population sizes and using a $\log_{10}$-scale. Figures 11(b), 12(b), 13(b), and 14(b) plot the corresponding median number of objective function evaluations taken. One can see that the pure version of CMA-ES behaves slightly better, when accurately searching



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

Figure 8: Performance profiles computed for the set of nonstochastic noisy problems with a logarithmic scale, considering the two levels of accuracy, 10^{-2} and 10^{-4}.



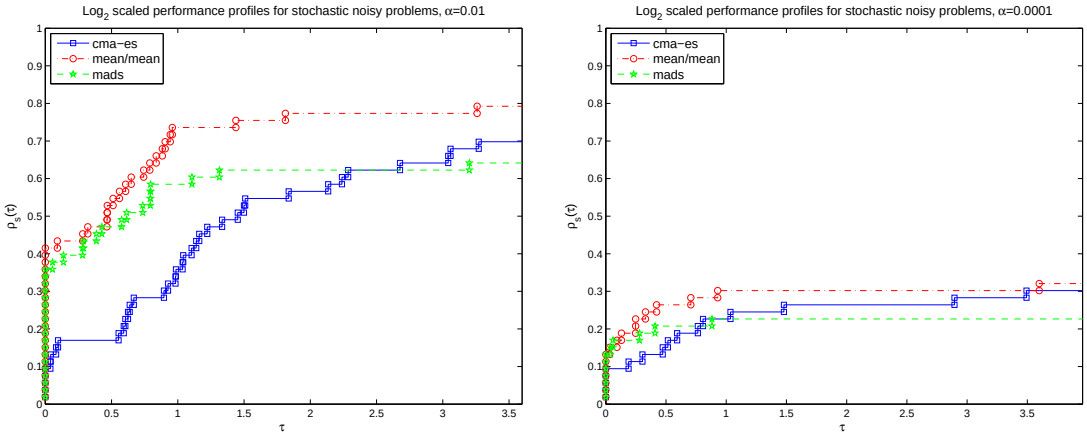
(a) Accuracy level of 10^{-2}. (b) Accuracy level of 10^{-4}.

Figure 9: Performance profiles computed for the set of piecewise smooth problems with a logarithmic scale, considering the two levels of accuracy, 10^{-2} and 10^{-4}.

for a global minimizer, in particular if a larger population size is given. The two approaches, however, exhibit difficulties in identifying a global minimizer in most of the problems within the given budget. The difficulty of this test set in terms of global optimization calls perhaps for additional algorithmic features such as a multistart technique.

6 Conclusions and Future Work

We have shown that it is possible to modify ES's so that they converge to stationary points without any assumption on the starting mean. The modified versions of the ES's promote smaller steps when the larger steps are uphill and thus lead to an improvement in the efficiency of the algorithms in the search of a stationary point. The so-called mean/mean version, where



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

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

Figure 10: Performance profiles computed for the set of stochastic noisy problems with a logarithmic scale, considering the two levels of accuracy, 10^{-2} and 10^{-4}.

Problem Number	11	12	13	14	15	16	17	18	19
Problem index in [18]	f_{122}	f_{123}	f_{124}	f_{125}	f_{126}	f_{127}	f_{128}	f_{129}	f_{130}

Table 3: Noisy problems.

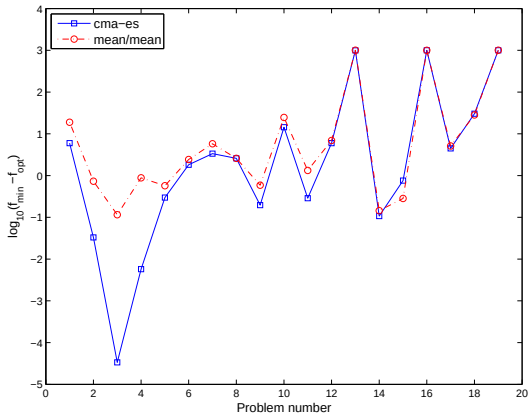
the step is reduced whenever the objective value of the weighted mean of the best trial offspring does not sufficiently reduce the objective value at the current weighted mean, has emerged as the best modified version in our numerical experiments. Apparently, the promotion of such smaller steps has not changed too much the search for the global minimizer in problems with several local minimizers.

Our approach applies to all ES's of the type considered in this paper (see Section 2) although we only used CMA-ES in our numerical tests. A number of issues regarding the interplay of our modifications in ES's (essentially the step-size update based on different sufficient decrease conditions) and the CMA scheme to update the covariance matrix and corresponding step size must be better understood and investigated. In addition, we have not explored to our benefit any hidden ability of the CMA scheme to approximate or predict first or second order information (which might be used in the sufficient decrease conditions or to guide the offspring generation).

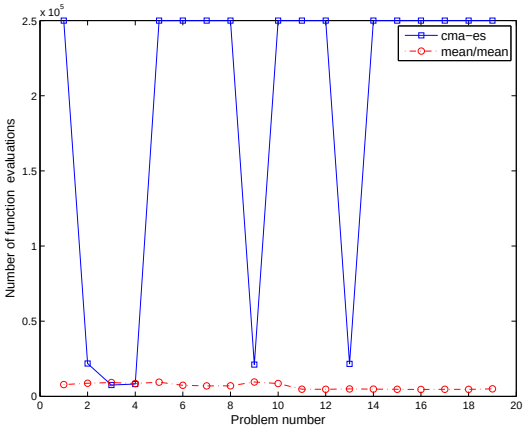
It is possible to significantly improve the numerical performance of ES's by incorporating a search step at the beginning of each iteration (as in the search-poll framework of direct search [7]). In such a step, one can, for instance, build a quadratic model using all or some of the points where the objective function has been previously evaluated and then minimize such a model in a certain region (see [11]). The application of such search steps to ES's as well as the extension to the constrained setting will be addressed in a forthcoming paper.

Acknowledgments

We would like to thank three anonymous referees, the associate editor, and the co-editor (Sven Leyffer) for their comments which improved the presentation of the paper.

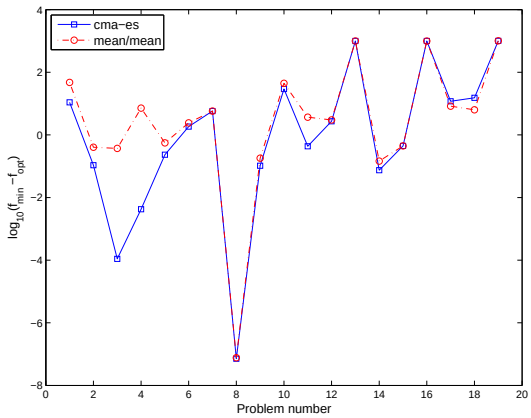


(a) Best function values (median).

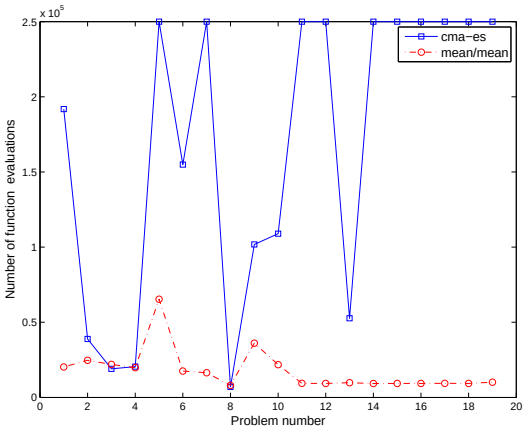


(b) Number of function evaluations taken (median).

Figure 11: Results for the mean/mean version and CMA-ES on a set of multi-modal functions of dimension 10 (using $\lambda = 20$).



(a) Best function values (median).



(b) Number of function evaluations taken (median).

Figure 12: Results for the mean/mean version and CMA-ES on a set of multi-modal functions of dimension 20 (using $\lambda = 40$).

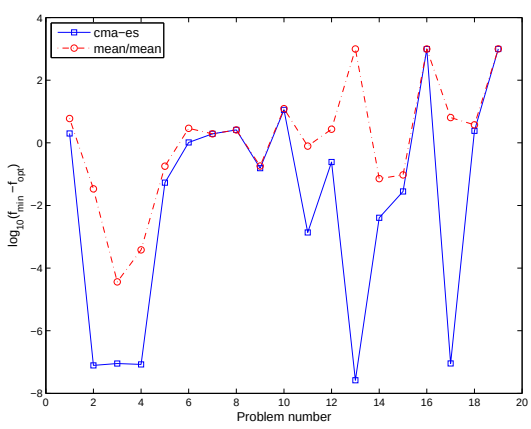
References

[1] 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`.

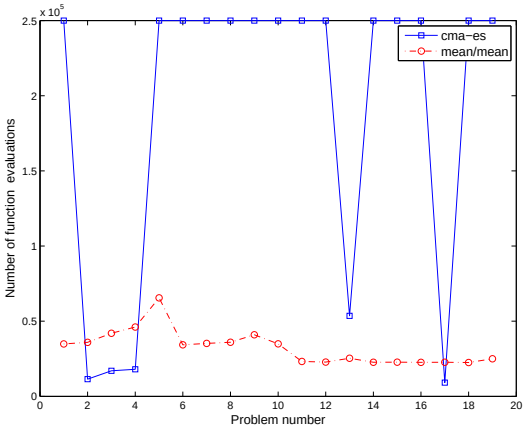
[2] C. Audet, S. Le Digabel, and C. Tribes. NOMAD user guide. Technical Report G-2009-37, Les cahiers du GERAD, 2009.

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

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

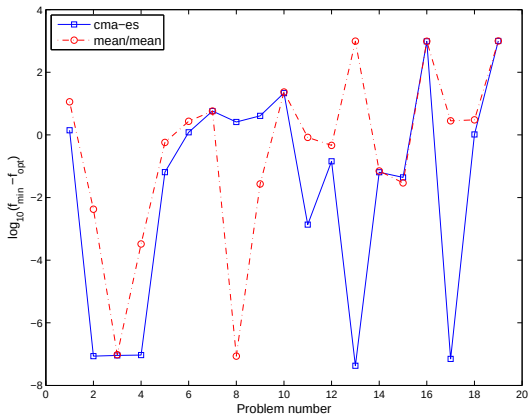


(a) Best function values (median).

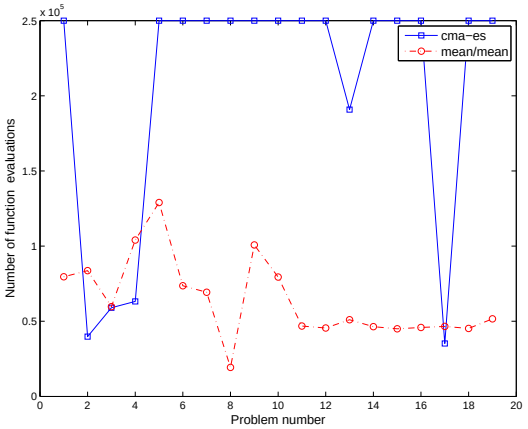


(b) Number of function evaluations taken (median).

Figure 13: Results for the mean/mean version and CMA-ES on a set of multi-modal functions of dimension 10 (using $\lambda = 100$).



(a) Best function values (median).



(b) Number of function evaluations taken (median).

Figure 14: Results for the mean/mean version and CMA-ES on a set of multi-modal functions of dimension 20 (using $\lambda = 200$).

[5] A. Auger. Convergence results for the (1, lambda)-SA-ES using the theory of phi-irreducible Markov chains. *Theoret. Comput. Sci.*, 334:35–69, 2005.

[6] H.-G. Beyer and H.-P. Schwefel. Evolution strategies: A comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

[7] 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.

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

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

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

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

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

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

[14] N. I. M. Gould, D. Orban, and P. L. Toint. CUTEr, a Constrained and Unconstrained Testing Environment, revisited. *ACM Trans. Math. Software*, 29:373–394, 2003.

[15] G. W. Greenwood and Q. J. Zhu. Convergence in evolutionary programs with self-adaptation. *Evolutionary Computation*, 9:57–147, 2001.

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

[17] N. Hansen, D. V. Arnold, and A. Auger. Evolution strategies. In J. Kacprzyk and W. Pedrycz, editors, *Handbook of Computational Intelligence*. Springer, Berlin, 2014, to appear.

[18] N. Hansen, S. Fincky, R. Rosz, and A. Auger. Real-parameter black-box optimization benchmarking 2010: Noisy functions definitions. Technical report, March 22, 2010.

[19] N. Hansen, S. Fincky, R. Rosz, and A. Auger. Real-parameter black-box optimization benchmarking 2010: Noiseless functions definitions. Technical report, September 28, 2010.

[20] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 312–317, 1996.

[21] 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.

[22] J. Jägersküpper. How the (1+1)-ES using isotropic mutations minimizes positive definite quadratic forms. *Theoret. Comput. Sci.*, 361:38–56, 2006.

[23] J. Jägersküpper. Probabilistic runtime analysis of (1+1)-ES using isotropic mutations. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 461–468, New York, NY, USA, 2006. ACM.

[24] 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.

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

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

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

[28] 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.

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