



HAL
open science

Injecting External Solutions Into CMA-ES

Nikolaus Hansen

► **To cite this version:**

Nikolaus Hansen. Injecting External Solutions Into CMA-ES. [Research Report] RR-7748, INRIA. 2011. inria-00628254

HAL Id: inria-00628254

<https://inria.hal.science/inria-00628254>

Submitted on 19 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Injecting External Solutions Into CMA-ES

Nikolaus Hansen

N° 7748

October 2011

Optimization, Learning and Statistical Methods



*R*apport
de recherche

Injecting External Solutions Into CMA-ES

Nikolaus Hansen

Theme : Optimization, Learning and Statistical Methods
Applied Mathematics, Computation and Simulation
Équipe-Projet TAO

Rapport de recherche n° 7748 — October 2011 — 12 pages

Abstract: This report considers how to inject external candidate solutions into the CMA-ES algorithm. The injected solutions might stem from a gradient or a Newton step, a surrogate model optimizer or any other oracle or search mechanism. They can also be the result of a repair mechanism, for example to render infeasible solutions feasible. Only small modifications to the CMA-ES are necessary to turn injection into a reliable and effective method: too long steps need to be tightly renormalized. The main objective of this report is to reveal this simple mechanism.

Depending on the source of the injected solutions, interesting variants of CMA-ES arise. When the best-ever solution is always (re-)injected, an elitist variant of CMA-ES with weighted multi-recombination arises. When *all* solutions are injected from an *external* source, the resulting algorithm might be viewed as *adaptive encoding* with step-size control.

In first experiments, injected solutions of very good quality lead to a convergence speed twice as fast as on the (simple) sphere function without injection. This means that we observe an impressive speed-up on otherwise difficult to solve functions. Single bad injected solutions on the other hand do no significant harm.

Key-words: CMA-ES, external solutions, gradient, injection, repair

Résumé : Pas de résumé

Mots-clés : Pas de motclef

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 3 |
| 2 | Injection in the CMA-ES Algorithm | 4 |
| 3 | Discussion | 7 |
| 4 | Preliminary Experiments | 9 |
| 5 | Further Considerations | 9 |
| 6 | Summary and Conclusion | 9 |

1 Introduction

The CMA-ES (Covariance Matrix Adaptation Evolution Strategy [4, 3, 2]) is a search stochastic algorithm for non-convex continuous optimization in a black-box setting, where we want minimize the objective function (or fitness function)

$$f : \mathbb{R}^n \rightarrow \mathbb{R}, \mathbf{x} \mapsto f(\mathbf{x})$$

without exploiting any a priori specified structure of f . The CMA-ES algorithm entertains a multivariate normal sampling distribution for \mathbf{x} and updates the distribution parameters with a comparatively sophisticated procedure, see Figure 1. While the algorithm is quite robust to large irregularities in the objective function f , even small changes of the update procedure can lead to a dramatic break down of its performance. This property has been perceived as a main weakness of the algorithm.

In this report we show how to make CMA-ES robust to (almost) arbitrary changes of the solutions used in the update procedure. In other words, we reveal the measures to properly inject external proposals for either candidate solution points or directions into the CMA-ES algorithm by replacing some of the *internal* solutions originally sampled by CMA-ES, or, equivalently, use solutions that are modified in any desired way (for example to make them feasible).

External or modified proposal solutions or directions can have a variety of sources.

- a gradient or Newton direction;
- an improved solution, for example the result of a local search started from a solution sampled by CMA-ES (Lamarckian learning), which allows to use CMA-ES in the context of memetic algorithms;
- a repaired solution, for example from a previously infeasible solution;
- an optimal solution of a surrogate model built from already evaluated solutions;
- the best-ever solution seen so far;
- proposals from any algorithm running in parallel to CMA-ES (migration).

Because injecting a single bad solution essentially corresponds to decreasing the population size by one, no particular care needs to be taken that *only* (exceptionally) good solutions are introduced. Any promising source of solutions might be used. Within CMA-ES, solutions are sampled symmetrically and therefore also virtually never lead to a systematic improvement before selection.

When all originally sampled internal solutions are replaced, the resulting procedure resembles *adaptive encoding* [6]. The main differences to adaptive encoding are: (i) external solutions are represented in the original (phenotypic) space (ii) step-size control remains in place and (iii) the parameter setting is different. Using a different (genotyp) representation to *generate* new external solutions is the crucial idea of adaptive encoding and can also be employed here.

The modifications introduced in CMA-ES are small but will often be decisive. They are outlined in the next section.

Notations Throughout this report, we use for $\mathbf{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\| = \sqrt{2}\Gamma(\frac{n+1}{2})/\Gamma(\frac{n}{2})$ the approximation $\sqrt{n}(1 - \frac{1}{4n} + \frac{1}{21n^2})$. The notation $a \wedge bc + d$ denotes the minimum of a and $bc + d$.

2 Injection in the CMA-ES Algorithm

The CMA-ES algorithm that tolerates injected solutions is displayed in Fig. 1. New parts are highlighted with shaded background. Injected solutions replace \mathbf{x}_i in (1). The decisive function $\alpha_{\text{clip}}(\cdot, \cdot)$ used in (3) and (6) reads

$$\alpha_{\text{clip}}(c, x) = 1 \wedge \frac{c}{x}. \quad (12)$$

However, different choices for $\alpha_{\text{clip}}(\cdot, \cdot)$ are possible, or even desirable, and discussed below. With parameter setting $c_y = c_y^{\text{m}} = \Delta_{\sigma}^{\text{max}} = \infty$, the original CMA-ES is recovered (in this case, Equations (3) and (6) are meaningless).

An injected direction, \mathbf{v} , is used by setting

$$\mathbf{x}_i = \mathbf{m}^t + \sigma^t \frac{\sqrt{n}}{\|\mathbf{C}^{-1/2}\mathbf{v}\|} \mathbf{v}. \quad (13)$$

If \mathbf{v} represents a gradient direction, using instead

$$\mathbf{x}_i = \mathbf{m}^t + \sigma^t \frac{\sqrt{n}}{\|\mathbf{C}^{1/2}\mathbf{v}\|} \mathbf{C}\mathbf{v} \quad (14)$$

seems to suggest itself. Remark that internal perturbations in CMA-ES follow $\mathbf{C}^{1/2}\mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is isotropic and $\|\mathcal{N}(\mathbf{0}, \mathbf{I})\| \approx \sqrt{n}$.¹

The decisive operation for injected solutions is given in Equation (3) of Figure 1. Their Mahalanobis distance to the distribution mean is clipped at $c_y \approx \sqrt{n} + 2$, preventing artificially long steps to enter the adaptation procedure. Additionally, but in most cases rather irrelevant after clipping the single steps, $\Delta_{\sigma}^{\text{max}}$ (see Table 1) keeps possible step-size changes below the factor $\exp(0.6) \approx 1.82$. Otherwise, the depicted algorithm is not further modified (unless c_y^{m} is set

¹The *symmetric* Cholesky factor $\mathbf{C}^{1/2}$ does not supply a *rotated* coordinate system as desired for adaptive encoding. In this case, we sample using $\mathbf{BD}\mathcal{N}(\mathbf{0}, \mathbf{I}) \sim \mathbf{C}^{1/2}\mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathbf{BD} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the linear decoding.

$$\begin{aligned} \mathbf{x}_i &\sim \mathbf{m}^t + \sigma^t \times \mathbf{C}^{t1/2} \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \text{for } i = 1, \dots, \lambda & (1) \\ \mathbf{y}_i &= \frac{\mathbf{x}_{i:\lambda} - \mathbf{m}^t}{\sigma^t} \quad \text{where } f(\mathbf{x}_{1:\lambda}) \leq \dots \leq f(\mathbf{x}_{\mu:\lambda}) \leq f(\mathbf{x}_{\mu+1:\lambda}) \dots & (2) \\ \mathbf{y}_i &\leftarrow \alpha_{\text{clip}}\left(c_y, \|\mathbf{C}^{t-\frac{1}{2}} \mathbf{y}_i\|\right) \times \mathbf{y}_i \quad \text{if } \mathbf{x}_{i:\lambda} \text{ was injected} & (3) \\ \Delta \mathbf{m} &= \begin{cases} \frac{\mathbf{x}_m - \mathbf{m}^t}{\sigma^t} & \text{if } \mathbf{x}_m \text{ was injected} \\ \sum_{i=1}^{\mu} w_i \mathbf{y}_i & \text{otherwise} \end{cases} & (4) \\ \mathbf{m}^{t+1} &= \mathbf{m}^t + c_m \sigma^t \Delta \mathbf{m} & (5) \\ \Delta \mathbf{m} &= \alpha_{\text{clip}}\left(c_y^m, \sqrt{\mu_w} \|\mathbf{C}^{t-\frac{1}{2}} \Delta \mathbf{m}\|\right) \times \Delta \mathbf{m} \quad \text{if } \mathbf{x}_m \text{ was injected or } \dots & (6) \\ \mathbf{p}_{\sigma}^{t+1} &= (1 - c_{\sigma}) \mathbf{p}_{\sigma}^t + \sqrt{c_{\sigma}(2 - c_{\sigma})\mu_w} \mathbf{C}^{t-\frac{1}{2}} \Delta \mathbf{m} & (7) \\ h_{\sigma} &= \begin{cases} 1 & \text{if } \|\mathbf{p}_{\sigma}^{t+1}\|^2 < n(1 - (1 - c_{\sigma})^{2(t+1)})(2 + 4/(n + 1)) \\ 0 & \text{otherwise} \end{cases} & (8) \\ \mathbf{p}_c^{t+1} &= (1 - c_c) \mathbf{p}_c^t + h_{\sigma} \sqrt{c_c(2 - c_c)\mu_w} \Delta \mathbf{m} & (9) \\ \mathbf{C}^{t+1} &= (1 - c'_1 - c_{\mu}) \mathbf{C}^t + c_1 \underbrace{\mathbf{p}_c^{t+1} \mathbf{p}_c^{t+1 \text{T}}}_{\text{rank one update}} + c_{\mu} \sum_{i=1}^{\mu} w_i \mathbf{y}_i \mathbf{y}_i^{\text{T}} & (10) \\ \sigma^{t+1} &= \sigma^t \times \exp\left(\Delta_{\sigma}^{\max} \wedge \frac{c_{\sigma}}{d_{\sigma}} \left(\frac{\|\mathbf{p}_{\sigma}^{t+1}\|}{\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1\right)\right) & (11) \end{aligned}$$

Figure 1: Update equations for the state variables in the $(\mu/\mu_w, \lambda)$ -CMA-ES with iteration index $t = 0, 1, 2, \dots$ and $\mathbf{m}^t \in \mathbb{R}^n$, $\sigma^t \in \mathbb{R}_+$, $\mathbf{C}^t \in \mathbb{R}^{n \times n}$ positive definite, $\mathbf{p}_{\sigma}^t, \mathbf{p}_c^t \in \mathbb{R}^n$ and $\mathbf{p}_{\sigma}^{t=0} = \mathbf{p}_c^{t=0} = \mathbf{0}$, $\mathbf{C}^{t=0} = \mathbf{I}$ and parameters taken from Table 1. We have additionally $c'_1 = c_1(1 - (1 - h_{\sigma}^2)c_c(2 - c_c))$. The chosen ordering of equations allows to remove the time index. The symbol $\mathbf{x}_{i:\lambda}$ is the i -th best of the solutions $\mathbf{x}_1, \dots, \mathbf{x}_{\lambda}$. The ‘‘optimal’’ ordering of (3), (4), (5) and (6) is an open issue.

$< \infty$). We also use the original internal strategy parameters for CMA-ES which seems particularly reasonable if only a smaller fraction of internal solutions is replaced in (1).

Strong injection: mean shift If we want to make a strong impact with an injection, we can shift the mean. We compute

$$\Delta \mathbf{m} = \frac{\mathbf{x}_m - \mathbf{m}^t}{\sigma^t} \quad (15)$$

from the injected solution \mathbf{x}_m as in (4). When no further solutions \mathbf{x}_i are used, the remaining update equations can be performed with $c_{\mu} = 0$. With $c_m = 1$ (the default), $\mathbf{m}^{t+1} = \mathbf{x}_m$. In order to prevent an unrealistic large shift of \mathbf{m}^t

Table 1: Default parameter values of $(\mu/\mu_w, \lambda)$ -CMA-ES taken from [5], where by definition $\sum_{i=1}^{\mu} |w_i| = 1$ and $\mu_w^{-1} = \sum_{i=1}^{\mu} w_i^2$ and $a - b \wedge c + d := \min(a - b, c + d)$. Only population size λ is possibly left to the users choice (see also [1])

| | |
|------------------------------|--|
| λ | $= 4 + \lfloor 3 \ln n \rfloor$ |
| μ | $= \lfloor \frac{\lambda}{2} \rfloor$ |
| w_i | $= \frac{\ln(\frac{\lambda+1}{2}) - \ln i}{\sum_{j=1}^{\mu} (\ln(\frac{\lambda+1}{2}) - \ln j)}$ |
| c_y | $= \sqrt{n} + 2n/(n+2)$ |
| c_y^m | $= \sqrt{2n} + 2n/(n+2)$ |
| c_m | $= 1$ |
| c_σ | $= \frac{\mu_w + 2}{n + \mu_w + 3}$ |
| d_σ | $= 1 + c_\sigma + 2 \max\left(0, \sqrt{\frac{\mu_w - 1}{n+1}} - 1\right)$ |
| c_c | $= \frac{4 + 0 \times \mu_w/n}{n + 4 + 0 \times 2\mu_w/n}$ |
| c_1 | $= \frac{\alpha_{\text{cov}} \min(1, \lambda/6)}{(n+1.3)^2 + \mu_w}$ |
| c_μ | $= 1 - c_1 \wedge \alpha_{\text{cov}} \frac{\mu_w - 2 + 1/\mu_w}{(n+2)^2 + \alpha_{\text{cov}} \mu_w/2}$ |
| α_{cov} | $= 2$ could be chosen < 2 , e.g. $\alpha_{\text{cov}} = 0.5$ for noisy problems |
| $\Delta_\sigma^{\text{max}}$ | $= 1.0$ or even 0.6 |

in (5) we might exchange the order of (5) and (6), therefore applying the length adjustment for $\Delta \mathbf{m}$ in (6) before the actual mean shift (5).

Parameter setting The setting of $c_y \approx \sqrt{n} + 2$ is motivated in Figure 2. The figure depicts the relative deviation of $\|\mathbf{C}^{-1/2} \mathbf{y}_i\|$ from its expected value. Given its original distribution from CMA-ES, less than 10% of the \mathbf{y}_i in (3) are actually clipped. For $n > 10$, the fraction is smaller than 1%.

The typical length of $\sqrt{\mu_w} \Delta \mathbf{m}$ depends on $\sqrt{\mu_w}$ and is often essentially larger than \sqrt{n} . Therefore the setting $c_y^m = \sqrt{n} + 2$ leads to a visible impairment of the otherwise unmodified CMA-ES. This suggests that $c_y^m \approx \sqrt{2n} + 2$ could be a reasonable choice, however the setting of c_y^m yet needs further empirical validation.

In principle, the **order of Equations** (3), (4), (5) and (6) can be changed under the constraint that the computation of $\Delta \mathbf{m}$ in (4) is done before $\Delta \mathbf{m}$ is used in (5) and (6). More specifically, four variants are available, implied by the exchange of (3) and (4), or (5) and (6), respectively, (another variant that uses unclipped \mathbf{y}_i for \mathbf{m}^{t+1} but clipped ones for $\Delta \mathbf{m}$ in the further computations is possible, however not by simple exchange of equations). The variants differ in whether $\Delta \mathbf{m}$ is computed from clipped \mathbf{y}_i and whether $\Delta \mathbf{m}$ itself is clipped before or after to compute \mathbf{m}^{t+1} . All these variations seem feasible, because an unconstrained shift of \mathbf{m}^{t+1} is per se not critical for the algorithm behavior.

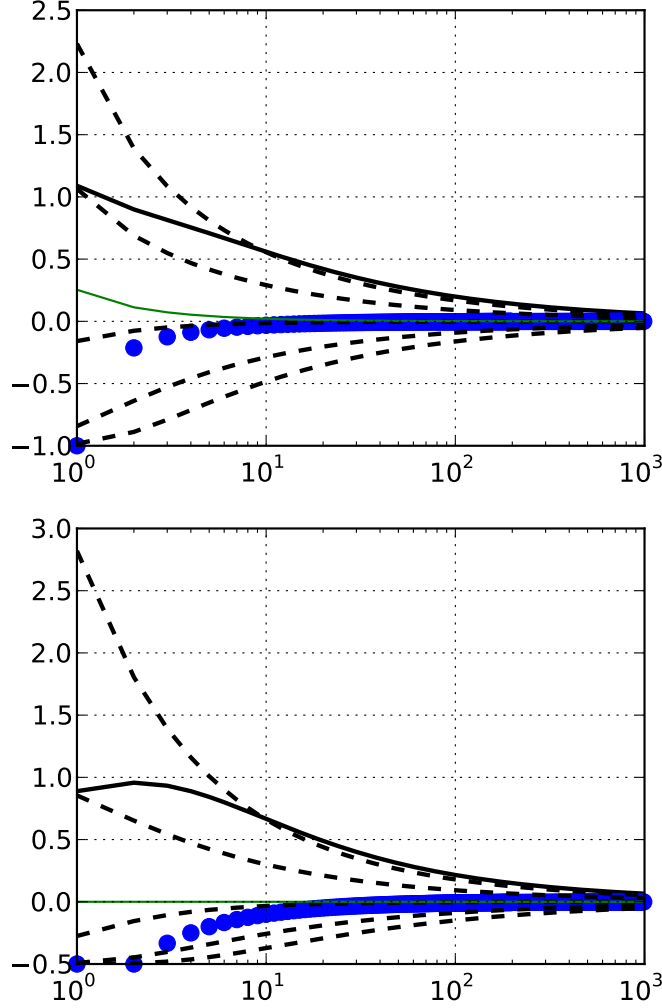


Figure 2: Relative deviation of $\|\mathbf{C}^{-1/2}\mathbf{y}_i\|$ (top) and $\|\mathbf{C}^{-1/2}\mathbf{y}_i\|^2$ (bottom) from its expected value plotted versus dimension. All values are normalized as $x \mapsto x/\mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\| - 1$ (top) and $x^2 \mapsto (x^2/n - 1)/2$ (bottom), compare also (11) for $c_\sigma = d_\sigma = 1$. Plotted are statistics of the random variable x (top) and x^2 (bottom), where x^2 follows a chi square distribution with n degrees of freedom, like $\|\mathbf{C}^{-1/2}\mathbf{y}_i\|^2$ does without injections under neutral selection. Plotted against n are modal value ($x = \sqrt{n-1}$ and $x^2 = 0 \vee n - 2$ respectively as dots), the approximation of the expected value \sqrt{n} and the expected value n respectively (thin solid), the 1, 10, 50, 90, and 99%tile (dashed) and $x = c_y = \sqrt{n} + 2n/(n+2)$.

3 Discussion

All update equations starting from (4) are formulated *relative* to the original sample distribution. This means we are, in principle, free to change the distri-

bution before each iteration step. Many reasonable adjustments are possible. A mean shift² (injecting \mathbf{x}_m resembles an arbitrary mean shift with additional further updates based on this mean shift), changing the step-size σ , increasing small variances in \mathbf{C} . . . The modification advised in this report is necessary, if \mathbf{x}_i is *not* in accordance with the distribution in (1).

With the introduced modification(s) the CMA-ES can also be used in the **adaptive encoding** context (however using σ for the encoding-decoding might only turn out to be useful if the encoding is an *affine* linear transformation). In the original adaptive encoding [6], different normalization measures have been taken for the cumulation in \mathbf{p}_c and for the covariance matrix update, and the step-size adaptation has been entirely omitted. In this report here, by default the tight normalization of the single steps is the only measure (unless \mathbf{x}_m is injected). The new normalization replaces the multiplication of the single steps with α_i in [6] in the covariance matrix update. The new normalization is tighter: choosing $c_y \approx 2\sqrt{n}$, instead of $c_y \approx \sqrt{n} + 2$, would be comparable to [6]. The setting therefore allows to apply step-size control reliably. However, the new setting is less tight for the mean step, as $c_y^m = \sqrt{n}$ (without taking a minimum in (6)) would be comparable to [6], while we use now $c_y^m = \infty$ unless an explicit mean-shift is performed. This setting might fail, if all new points point into the same direction viewed from \mathbf{x}_m (suggesting $c_y^m \approx 2\sqrt{n}$ as a compromise). The new setting seems to be slightly simpler and might turn out preferable also in the adaptive encoding setting, even when leaving aside step-size adaptation.

Due to the minor modifications we do not expect an adverse interference with **negative updates** of the covariance matrix as in active CMA-ES [7, 5]. On the contrary, limiting the length of steps that enter the negative update mitigates a principle design flaw of negative updates: long steps tend to be worse (and therefore enter the negative update with a higher probability) and tend to produce a stronger update effect, both just because they are long and not because they indicate an undesirable direction.

Finally, it is well possible to inject the same solution several times, for example based on its superior quality. One might, for example, consider to unconditionally (re-)inject the best-ever solution in every iteration. Then, an “elitist algorithm with comma selection” arises—introducing an easy and appealing way to **implement elitism** in evolution strategies with weighted multi-recombination.

A **generalized approach** to normalize injected solutions compares the empirical CDF of the lengths $l_i^t = \|\mathbf{C}^{t-\frac{1}{2}}\mathbf{y}_i\|$, $i = 1, \dots, \lambda$, $t = 1, 2, \dots$, with a desired CDF, F , and reduces the length of \mathbf{y}_i such that the observed relative frequency of lengths larger than or equal to l_i^t is below, say, $1.2(1 - F(l_i^t))$. In (12), the desired CDF is very crudely chosen to be $F(x) = 1_{x < c_y}$. The desired operation in theory is $l_i^t \leftarrow F_{\text{desired}}^{-1}(F_{\text{true}}(l_i^t))$. A practicable implementation could compare $l_i^t = \sqrt{2} \times (\|\mathbf{C}^{t-\frac{1}{2}}\mathbf{y}_i\| - \mathbb{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|)$ with the standard normal distribution \mathcal{F} , in that a correction is applied if $l_i^t > 1$ and $\frac{1}{t} \frac{1}{\lambda} \sum_{j,k} 1_{l_j^k \geq l_i^t} > 1.2(1 - \mathcal{F}(l_i^t))$.

²However a mean shift without further updates will impair the meaning of the evolution paths.

4 Preliminary Experiments

Preliminary empirical investigations have been conducted by injecting a single slightly perturbed *optimal* solution. This is virtually the best case scenario when the distribution mean \mathbf{m} is far away from the optimum. This becomes the worst case scenario, when \mathbf{m} is closer to the optimum than the perturbation. Single runs on the sphere function and the Rosenbrock function are shown in Figures 3 and 4. The upper black graph depicts the worst iteration-wise function value and reveals the (sharp) transition between best and worst case scenario by showing convergence first and stagnation afterwards.

The improvement on the sphere function is limited to a factor of about two, namely due to the maximal iteration-wise step-size decrement. This limit can be exceeded by additionally decreasing σ when the injected solution is trustworthy, has a good quality, and is close to \mathbf{m} (in the norm defined by $\sigma^2\mathbf{C}$). The precise implementation (the question of what is close to \mathbf{m}) might also depend on μ_w . As to be expected, the effect of injecting single bad solutions (worst case scenario in the later stage) is negligible.

The improvement on the Rosenbrock function exceeds our expectation: we see a speed-up by a factor of almost n , simply because the speed is similar to the one on the sphere function with injection. Again, this speed-up can be further enhanced by step-size decrements.

Experiments for an injected mean-shift have not been conducted yet.

Experiments injecting always the best-ever solution reveal a moderate performance impairment when searching multimodal landscapes.

5 Further Considerations

Another case of application is temporary freezing of some variables (coordinates) to the same value in all candidate solutions \mathbf{x}_i . (This decreases the length of the step in the Euclidean norm, but due to correlations in the distribution this can lead to exceptionally long steps in Mahalanobis distance even if the frozen value is borrowed from \mathbf{m}). In this case, it is also advisable to slightly modify the step-size equations (8) and (11). Given j variables are frozen, these variables are not taken into account for computing $\|\mathbf{p}_\sigma^{t+1}\|$ and consequently $n - j$ is used instead of n in (8) and $\mathbf{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|$ is computed for $n - j$ dimensions in (11). After one iteration, the respective components of \mathbf{y}_i will be zero (given $c_m = 1$) and also c_y should be set as for dimension $n - j$. In principle, all parameters from Table 1 can then be set as for dimension $n - j$. Additionally, in order to avoid numerical problems, the diagonal elements of the frozen coordinates of the covariance matrix should be kept at least in the order of the smallest eigenvalue.

6 Summary and Conclusion

Using candidate proposals in the CMA-ES that do not directly stem from the sample distribution of CMA-ES can often lead to a failure of the algorithm. The effective counter measures however turn out to be comparatively simple: only the appearance of large steps needs to be tightly controlled, where large is defined w.r.t. the original sample distribution. The possibility to inject any candidate solution is valuable in many situations. In case of bounds or constraints

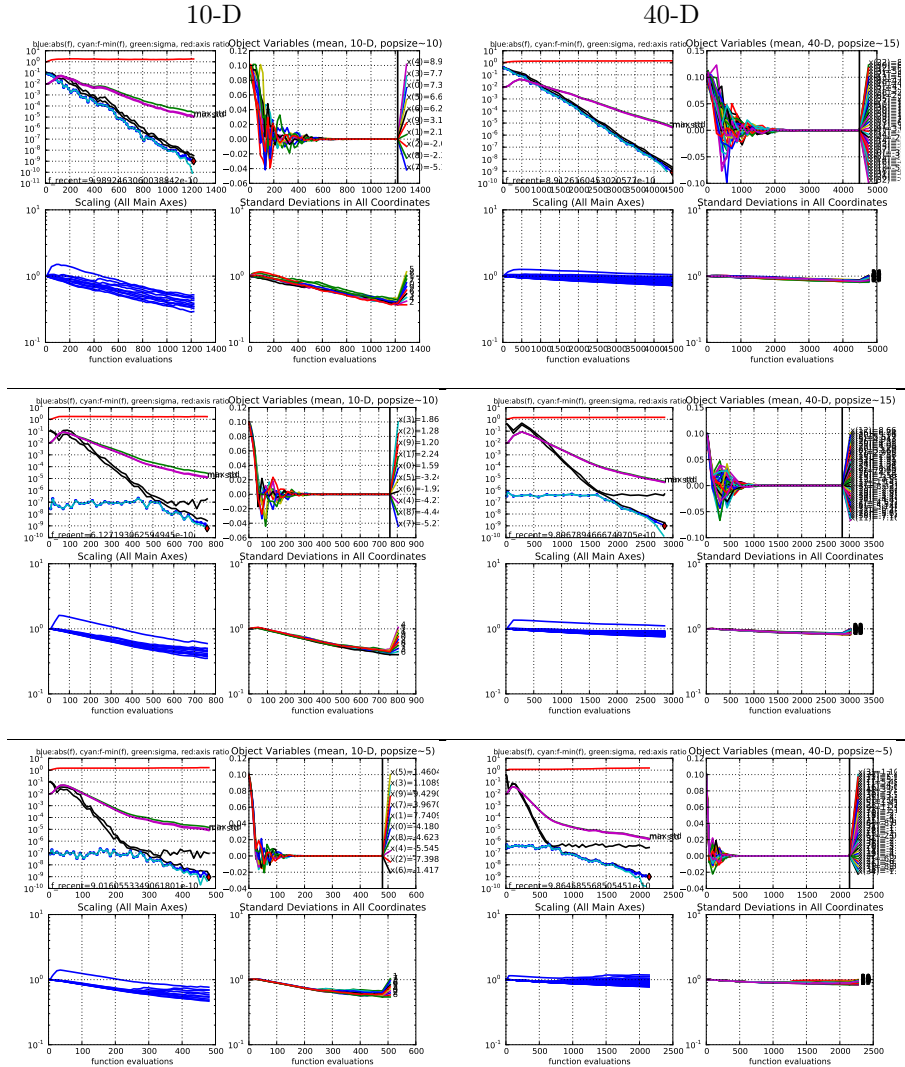


Figure 3: Runs of CMA-ES on the sphere function, middle and lower row with a single injected solution distributed as $10^{-4} \times \mathcal{N}(\mathbf{0}, \mathbf{I})$ (which corresponds in the beginning to a slightly disturbed gradient direction). Black lines show the evolution of median and worst solution. The evolution of the worst solution indicates that—with default parameter setting on the sphere function—a speed-up by a factor of two can be achieved. The reason for the comparatively moderate speed-up is that the step-size decrease per iteration is limited. With reduced population size (lower row) the speed-up increases because the number of iterations to reach function value 10^{-6} in the best case scenario remains almost constant.

where a repair mechanism is available, this might serve as basis for a new class of well-performing constraint handling mechanisms.

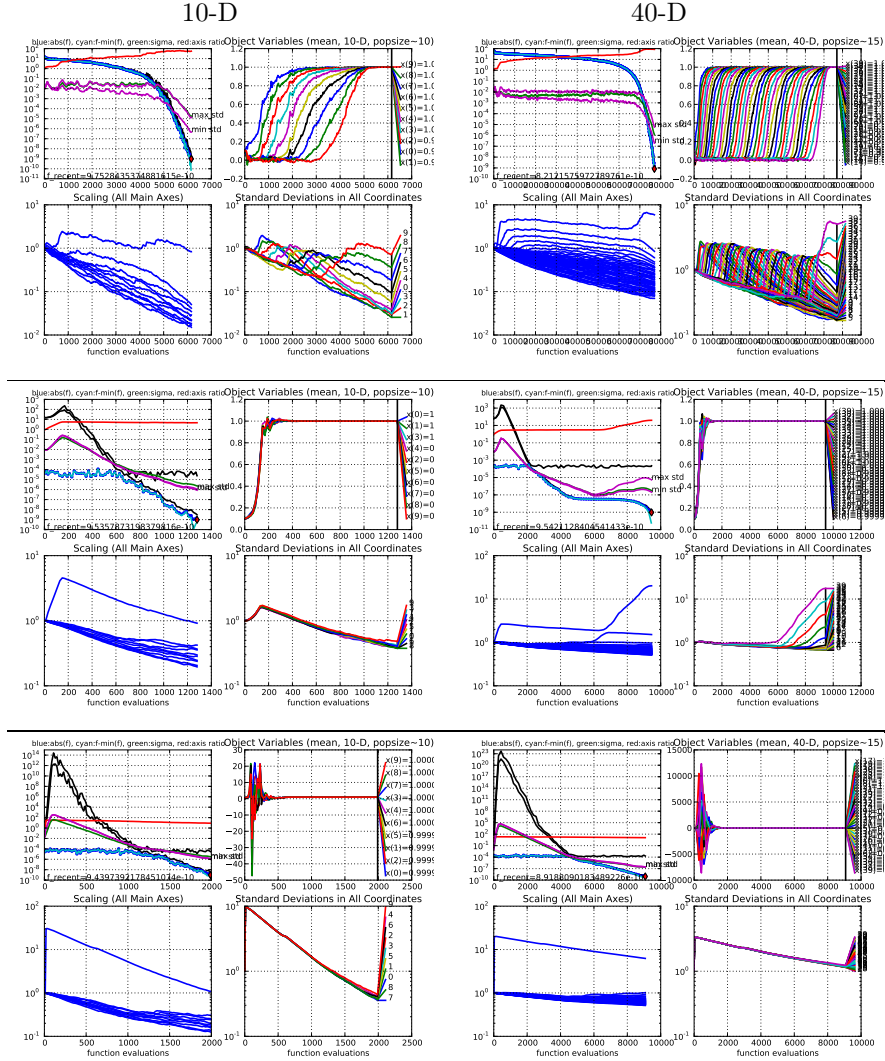


Figure 4: Runs of CMA-ES on the Rosenbrock function, middle and lower row with a single injected solution distributed as $\mathbf{1} + 10^{-4} \times \mathcal{N}(\mathbf{0}, \mathbf{I})$, lower row without the clipping in (3). With injection, the convergence speed is again twice as fast as on the sphere function. Where the injected solutions are useful (for function values larger than about 10^{-4}), the algorithm is almost n times faster than without injection (600 vs 5000 and 2000 vs 70000 evaluations in 10- and 40-D). Without clipping, the run does not fail only because the step-size increment is limited to $\exp(\Delta_{\sigma}^{\max}) = 2.718\dots$ per iteration.

Acknowledgment This work was supported by the ANR-2010-COSI-002 grant (SIMINOLE) of the French National Research Agency.

References

- [1] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proc. IEEE Congress On Evolutionary Computation*, pages 1769–1776, 2005.
- [2] N. Hansen and S. Kern. Evaluating the CMA Evolution Strategy on multimodal test functions. In X. Yao et al., editors, *Parallel Problem Solving from Nature PPSN VIII*, volume 3242 of *LNCS*, pages 282–291. Springer, 2004.
- [3] N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation. *Evolutionary Computation*, 11(1):1–18, 2003.
- [4] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [5] N. Hansen and R. Ros. Benchmarking a weighted negative covariance matrix update on the BBOB-2010 noiseless testbed. In *GECCO 2010 Proceedings of the 12th annual conference companion on Genetic and evolutionary computation Genetic And Evolutionary Computation Conference*, pages 1673–1680, Portland United States, 2010.
- [6] Nikolaus Hansen. Adaptive encoding: How to render search coordinate system invariant. In G. Rudolph et al., editors, *Parallel Problem Solving from Nature (PPSN X)*, LNCS, pages 205–214, 2008.
- [7] G.A. Jastrebski and D.V. Arnold. Improving evolution strategies through active covariance matrix adaptation. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 2814–2821, 2006.



Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399