HAL
open science

CMA-ES with Two-Point Step-Size Adaptation

Nikolaus Hansen

To cite this version:

Nikolaus Hansen. CMA-ES with Two-Point Step-Size Adaptation. [Research Report] RR-6527, INRIA. 2008. inria-00276854v5

HAL Id: inria-00276854

https://inria.hal.science/inria-00276854v5

Submitted on 18 May 2008

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.

inria-00276854, version 5 - 18 May 2008

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

CMA-ES with Two-Point Step-Size Adaptation

Nikolaus Hansen

N° 6527

May 2008

Thème COG

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6527--FR+ENG

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche SACLAY - ÎLE-DE-FRANCE

CMA-ES with Two-Point Step-Size Adaptation

Nikolaus Hansen*

Thème COG — Systèmes cognitifs
Équipes-Projets Adaptive Combinatorial Search et TAO

Rapport de recherche n° 6527 — May 2008 — 9 pages

Abstract: We combine a refined version of *two-point step-size adaptation* with the *covariance matrix adaptation evolution strategy* (CMA-ES). Additionally, we suggest polished formulae for the learning rate of the covariance matrix and the recombination weights. In contrast to *cumulative step-size adaptation* or to the *1/5-th success rule*, the refined two-point adaptation (TPA) does not rely on any internal model of optimality. In contrast to conventional *self-adaptation*, the TPA will achieve a better target step-size in particular with large populations. The disadvantage of TPA is that it relies on two additional objective function evaluations.

Key-words: optimization, evolutionary algorithms, covariance matrix adaptation, step-size control, self-adaptation, two-point adaptation

* Adaptive Combinatorial Search Team, Microsoft Research–INRIA Joint Centre. 28, rue Jean Rostand, 91893 Orsay Cedex, France. email:`forename.name@inria.fr`.

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

Adaptation du Pas Deux-Point dans CMA-ES

Résumé : Pas de résumé

Mots-clés : Pas de motclef

1 Introduction

In the Covariance Matrix Evolution Strategy (CMA-ES) [8] two separate adaptation mechanism are performed to determine variances and covariances of the search distribution. One for (overall) step-size control, a second for adaptation of a covariance matrix. The mechanisms are mainly independent and can therefore, in principle, be replaced separately. While the standard step-size control is *cumulative step-size adaptation* (CSA), also a success-based control was successfully introduced for the $(1+\lambda)$-CMA-ES in [9].

The CSA has a few drawbacks.

- For very large noise levels the target step-size becomes zero, while the optimal step-size is still positive [3].
- For large population sizes ($\lambda > 10\,n$) the original parameter setting seemed not to work properly [6]—the notion of tracking a (long) path history seems not to perfectly mate with a large population size (large compared to the search space dimension). An improved parameter setting introduced in [5] shortens the backward time horizon for the cumulation and performs well also with large population sizes [5, 2].
- The expected size for the displacement of the population mean under random selection is required. To compute a useful measurement independent of the coordinate system, the principle axes of the search distribution are needed. They are more expensive to acquire (at least by a constant factor) than a simple matrix decomposition that is in any case necessary to sample a multivariate normal distribution with given covariance matrix.
- Because the length of an evolution path is compared to its expected length, the measurement is sensitive to the specific sample procedure of new candidate solutions and also, for example, to repair mechanisms for solutions.

Despite these disadvantages, CSA is regarded as first choice for step-size control in the $(\mu/\mu_{\mathrm{w}}, \lambda)$-ES, due to its advantages. Nonetheless, the disadvantages rise motivation to search for alternatives. Here, we suggest *two-point step-size adaptation* (TSA) as one such alternative.

Two-point self-adaptation was introduced for backpropagation in [11] and later applied in Evolutionary Gradient Search [10]. In evolutionary search, two-point adaptation resembles self-adaptation on the population level. The principle is utmost simple: two different step lengths are tested for the mean displacement and the better one is chosen. In the next section, we integrate a slightly refined TSA in the CMA-ES and additionally introduce polished formulae for the recombination weights and the learning rates of the covariance matrix.

2 The Algorithm: CMA-ES with TPA

Our description of the CMA-ES closely follows [4, 5, 7] and replaces CSA with TSA. Given an initial mean value $\boldsymbol{m} \in \mathbb{R}^n$, the initial covariance matrix $\boldsymbol{C} = \mathbf{I}$ and the initial step-size $\sigma \in \mathbb{R}_+$, the new candidate solutions $\boldsymbol{x}_k$ obey

$$\boldsymbol{x}_k = \boldsymbol{m} + \sigma\, \boldsymbol{y}_k, \quad \text{for} \quad k = 1, \ldots, \lambda \ , \tag{1}$$

where $\boldsymbol{y}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C})$ denotes the realization of a normally distributed random vector with zero mean and covariance matrix $\boldsymbol{C}$. The solutions $\boldsymbol{x}_k$ are evaluated and ranked such that $\boldsymbol{x}_{i:\lambda}$ becomes the i-th best solution vector and $\boldsymbol{y}_{i:\lambda}$ the corresponding random vector realization.

For $\mu < \lambda$ let

$$\langle \boldsymbol{y} \rangle = \sum_{i=1}^{\mu} w_i \boldsymbol{y}_{i:\lambda}, \quad w_1 \geq \cdots \geq w_\mu > 0, \quad \sum_{i=1}^{\mu} w_i = 1 \tag{2}$$

be the weighted mean of the μ best ranked $\boldsymbol{y}_k$ vectors. The recombination weights sum to one. The *variance effective selection mass* is defined as

$$\mu_{\mathrm{w}} = \frac{\left(\sum_{i=1}^{\mu} w_i\right)^2}{\sum_{i=1}^{\mu} w_i^2} = \frac{1}{\sum_{i=1}^{\mu} w_i^2} \geq 1 \ . \tag{3}$$

From the definition follows that $1 \leq \mu_{\mathrm{w}} \leq \mu$ and $\mu_{\mathrm{w}} = \mu$ for equal recombination weights. The role of μ_{w} is analogous to the role of the parent number μ when the recombination weights are all equal. Usually $\mu_{\mathrm{w}} \approx \lambda/4$ is appropriate. Weighted recombination is discussed in more detail in [1].

The default parameter values are

$$\lambda = 4 + \lfloor 3 \ln n \rfloor, \quad \mu' = \frac{\lambda}{2}, \quad \mu = [\mu'] \quad \text{and} \tag{4}$$

$$w_i = \frac{\ln(\mu' + 0.5) - \ln i}{\sum_{j=1}^{\mu} (\ln(\mu' + 0.5) - \ln j)} \quad \text{for} \quad i = 1, \ldots, \mu \ , \tag{5}$$

where $[\mu']$ denotes the integer value closest to μ', preferably chosing the smaller integer value in case, such that $w_{[\mu']} > 0$. The first $[0.2\mu']$ weights sum to about 0.5. Conducting restarts with increasing value of λ is a valuable option [2].

In the remainder, the generation step is completed with the updates of $\boldsymbol{m}$, σ, and $\boldsymbol{C}$, where two additional state variables, $\alpha_{\mathrm{s}} \in \mathbb{R}$ and $\boldsymbol{p_c} \in \mathbb{R}^n$, will be introduced and the method parameters are discussed in Section 2.4.

2.1 The Mean

The distribution mean is updated according to

$$\boldsymbol{m} \leftarrow \boldsymbol{m} + \sigma \langle \boldsymbol{y} \rangle \ . \tag{6}$$

Given σ from Equation (1), Equation (6) can also be written as

$$\boldsymbol{m} \leftarrow \sum_{i=1}^{\mu} w_i \boldsymbol{x}_{i:\lambda} \ . \tag{7}$$

2.2 Step-Size Control: Two-Point Adaptation (TPA)

A two-point self-adaptive scheme is implemented based on [10]. We compute two additional function evaluations

$$f_+ = f(\boldsymbol{m} + \alpha' \, \sigma \, \langle \boldsymbol{y} \rangle) \tag{8}$$

$$f_- = f(\boldsymbol{m} - \alpha' \, \sigma \, \langle \boldsymbol{y} \rangle) \ , \tag{9}$$

where f is the objective function to be minimized, $\boldsymbol{m}$ is the new (updated) mean value, and $\alpha' \approx 0.5$ is the test width parameter. The factor $\pm\alpha'$ in the equations is chosen symmetrical about the new mean $\boldsymbol{m}$.

The step-size should increase if f_+ is better than f_-, and decrease otherwise. Using the values f_+ and f_- we set

$$\alpha_{\text{act}} = \begin{cases} -\alpha + \beta & < 0, \text{ if } f_- \text{ is better (smaller) than } f_+ \\ \alpha & > 0, \text{ otherwise} \end{cases} \tag{10}$$

Initializing $\alpha_{\text{s}} = 0$, the new step-size is calculated according to

$$\alpha_{\text{s}} \leftarrow \alpha_{\text{s}} + c_\alpha(\alpha_{\text{act}} - \alpha_{\text{s}}) = (1 - c_\alpha)\, \alpha_{\text{s}} + c_\alpha \alpha_{\text{act}} \tag{11}$$

$$\sigma \leftarrow \sigma \times \exp(\alpha_{\text{s}}) \tag{12}$$

where $1/c_\alpha \geq 1$ determines the backward time horizon for smoothing the step-size changes in the generation sequence. The default parameter settings are

$$\alpha' = 0.5, \quad \alpha = 0.5, \quad \beta = 0, \quad c_\alpha = 0.3 \ . \tag{13}$$

Comparison to the previous formulation The two-point step-size adaptation described here differs from [10] in that the test steps are distinguished from the step-size changes by using (i) a *symmetrical* test step about the new $\boldsymbol{m}$, (ii) different test width and change parameters and (iii) a smoothing for the step-size change. Furthermore, the *original* step-size is used for updating $\boldsymbol{m}$. Setting $\alpha' = 0.8$, $\alpha = \ln(1.8) \approx 0.588$, $\beta = 0$, $c_\alpha = 1$, replacing $-\alpha'$ with $-\alpha'/(1 + \alpha')$ in Equation (9) and using the new step-size for finally updating the mean $\boldsymbol{m}$ recovers the step-size adaptation from [10]. We do not expect an essentially different behavior due to our refinements in most cases.

Step-size changes are essentially multiplicative. A factor $\exp(\pm\alpha)$ can be used to realize changes of σ, which is symmetrical about 1 in the log scale. On the other hand, using such factors for generating *test* steps extends the step further by $\exp(+\alpha) > 1$ than reducing it by $\exp(-\alpha) < 1$. Assuming the

most simple spherical objective function model and *optimal step-size*, where $f(\boldsymbol{m} + a\langle\boldsymbol{y}\rangle)$ about the new mean $\boldsymbol{m}$ is minimal for $a = 0$ and

$$f(\boldsymbol{m} + a\langle\boldsymbol{y}\rangle) = f(\boldsymbol{m} - a\langle\boldsymbol{y}\rangle) \ ,$$

a larger test step

$$f(\boldsymbol{m}_{\mathrm{old}} + \exp(\alpha'')\sigma\langle\boldsymbol{y}\rangle) = f(\boldsymbol{m} + \alpha'\sigma\langle\boldsymbol{y}\rangle) \ ,$$

given $\alpha'' = \ln(1 + \alpha')$, is disfavored compared to

$$f(\boldsymbol{m}_{\mathrm{old}} + \exp(-\alpha'')\sigma\langle\boldsymbol{y}\rangle) = f\Big(\boldsymbol{m} - \frac{\alpha'}{1+\alpha'}\sigma\langle\boldsymbol{y}\rangle\Big) \ .$$

The step-size will systematically decrease, the target step-size is smaller than the optimal step-size. On simple functions, like the sphere model, this effect might well lead to a performance improvement, because the optimum can be approached quickly and therefore the optimal step-size decreases fast. The sub-optimal target step-size "anticipates" this change. Nevertheless, in general, we tend to favor an agreement of target and optimal step-size and therefore we are in favor of symmetrical test steps.[1]

Following [10], the update of $\boldsymbol{m}$ in Equation (6) could be postponed until after the step-size is updated in Equation (12) (Equations (8) and (9) must be revised accordingly using the old mean value). Whether or not this results in a better $\boldsymbol{m}$ cannot be decided without additional costs, because neither the original step-size nor the updated step-size are usually tested. Furthermore, Equation (7) would not hold anymore. Empirically, using the new step-size leads to slightly higher convergence rates in norm optimization (sphere function) in small dimensions.

2.3 Covariance Matrix Adaptation (CMA)

The covariance matrix admits a rank-one and a rank-μ update. For the rank-one update an evolution path $\boldsymbol{p_c}$ is constructed.

$$\boldsymbol{p_c} \leftarrow (1 - c_{\boldsymbol{c}})\,\boldsymbol{p_c} + h_\sigma\sqrt{c_{\boldsymbol{c}}(2 - c_{\boldsymbol{c}})\mu_{\mathrm{W}}}\,\langle\boldsymbol{y}\rangle \tag{14}$$

$$\boldsymbol{C} \leftarrow (1 - c_1 - c_\mu)\,\boldsymbol{C} + c_1 \underbrace{\boldsymbol{p_c}\boldsymbol{p_c}^{\mathrm{T}}}_{\text{rank-one update}} + c_\mu \underbrace{\sum_{i=1}^{\mu} w_i \boldsymbol{y}_{i:\lambda}\boldsymbol{y}_{i:\lambda}^{\mathrm{T}}}_{\text{rank-}\mu\text{ update}} \ , \tag{15}$$

where $h_\sigma = 0$ if $\alpha_{\mathrm{s}} > (1 - (1 - c_\alpha)^9)(1 - (1 - c_\alpha)^g)\,\alpha$, and 1 otherwise, where g is the generation counter. The update of $\boldsymbol{p_c}$ is stalled when α_{s} is large. The stall is decisive after a change in the environment which demands a significant increase of the step-size. Fast changes of the distribution shape are postponed until after the step-size has increased to a reasonable value.

[1] Good algorithm design must at times prefer the reasonable to the optimal performance in order to avoid overfitting to specific test scenarios.

For the covariance matrix update, the cumulation in (14) serves to capture dependencies between consecutive steps. Dependency information would be lost for $c_{\boldsymbol{c}} = 1$, because a change in sign of $\boldsymbol{p_c}$ or $\boldsymbol{y}_{i:\lambda}$ does not matter in (15).

The default parameter settings are

$$c_{\boldsymbol{c}} = \frac{4}{n+4}, \quad \mu_{\mathrm{cov}} = \mu_{\mathrm{w}}, \tag{16}$$

$$c_1 = \frac{2}{(n+1.3)^2 + \mu_{\mathrm{cov}}}, \quad c_\mu = \min\left(2\,\frac{\mu_{\mathrm{cov}} - 2 + \frac{1}{\mu_{\mathrm{cov}}}}{(n+2)^2 + \mu_{\mathrm{cov}}}\,,\, 1 - c_1\right) \quad . \tag{17}$$

2.4 Discussion of Parameters

The default values for all parameters, namely offspring population size λ, recombination weights $w_{i=1,\ldots,\mu}$, cumulation parameter $c_{\boldsymbol{c}}$, mixing number μ_{cov}, and learning rates c_1 and c_μ follow [4, 5, 7] and were given above, as well as the step-size parameters test width α', changing factor α, update bias β and smoothing parameter c_α. The changes of parameters compared to [4, 5, 7] are minor polishings. We discuss some settings in detail.

Recombination weights Compared to [4, 5, 7], where $\mu' = \lceil(\lambda - 1)/2\rceil$ we have chosen $\mu' = (\lambda - 1)/2$. The small difference occurs only for even λ. In the former version, given odd population size λ, the recombination weights did not change when λ was reduced by one. In the present version the recombination weights always adjust to changes of λ.

c_1 **and** c_μ are the learning rates for the rank-one and rank-μ update of the covariance matrix respectively. In [4, 5, 6, 7], a learning rate $c_{\mathrm{cov}} \approx c_1 + c_\mu$ is used such that $c_1 \approx c_{\mathrm{cov}}/\mu_{\mathrm{cov}}$ and $c_\mu \approx c_{\mathrm{cov}}(\mu_{\mathrm{cov}} - 1)/\mu_{\mathrm{cov}}$. In the former formulation, c_1 was almost two times smaller for values of $\mu_{\mathrm{cov}} \approx 2$ than for $\mu_{\mathrm{cov}} = 1$ and did not monotonously decrease with larger μ_{cov}.

c_α determines the smoothing of α_{s}. Smoothing and choosing α small (damping) suppress stochastic fluctuations of σ. In contrast to choosing α small, smoothing does not affect the maximal possible change rate for σ. For $c_\alpha \geq 0.5$ we find $\alpha_{\mathrm{act}}\alpha_{\mathrm{s}} > 0$. Signs of the recent measurement and the actual change always agree and the smoothing cannot lead to oscillations. For $c_\alpha \geq 0.3$ only after a second agreeing measure for α_{act} we have *always* $\alpha_{\mathrm{act}}\alpha_{\mathrm{s}} > 0$. Even smaller values for c_α might be useful, but for much smaller values, presumably α must be chosen more carefully (smaller).

β is the bias parameter for the step-size change. On potentially noisy or highly rugged functions β should be set to $0.2\,\alpha$ which results in an effective noise handling.

3 Empirical Validation

In empirical investigations of the TPA-CMA-ES, we find the expected, feasible behavior. The comparison with CSA shows no clear winner. Depending on the objective function either TPA or CSA is faster, but the factor is seldom larger than two. Surprisingly, in our exploratory simulations, there is no clear winner depending on dimension or population size or noise. On noisy functions, setting $\beta = 0.2\,\alpha = 0.1$ for TPA is quite effective, while we observe only a minor effect from this change otherwise. We did not extensively try to exploit potential weaknesses (as has been done for CSA), but we suspect that the TPA is a feasible and robust alternative to CSA.

4 Conclusion and Outlook

We see some principle **advantages** of using two-point step-size adaptation (TPA) in the CMA-ES.

- The TPA does not rely on a predefined optimality condition, like a success rate of 1/5 or conjugate-perpendicularity of consecutive steps.
- The TPA does not rely on specific properties of the sample distribution or the selection of solutions. Therefore, it is supposably less sensitive to any modifications of the underlying algorithm, in particular compared to CSA.
- The step-size change rate can be adjusted mainly independently from TPA-internal considerations. Time averaging or damping are not essentially necessary.

Even so, we see two principle **disadvantages** of TPA.

- Two additional function evaluations are needed per iteration step. This is not a grave disadvantage, in particular when the population size is not very small. As a possible remedy, these two points could be incorporated in the population and used to compute the (final) mean in Equation (7), and one of them might be used in the rank-μ update of the covariance matrix.
- Step-size control is based on two objective function evaluations only. Selection information from the remaining population (and history information) is somewhat disregarded. This is a conceptional defect, that might be irrelevant in practice.

In conclusion, two-point step-size adaptation is an alternative to cumulative step-size adaptation well worth of further exploration. Whether and when it should finally replace CSA in practice must be answered in future empirical studies.

References

[1] D.V. Arnold. Weighted multirecombination evolution strategies. *Theoretical Computer Science*, 361(1):18–37, 2006.

[2] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 1769–1776. IEEE Press, 2005.

[3] H.-G. Beyer and D. Arnold. Qualms regarding the optimality of cumulative path length control in CSA/CMA-evolution strategies. *Evolutionary Computation*, 11(1):19–28, 2003.

[4] N. Hansen. The CMA evolution strategy: a comparing review. In J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer, 2006.

[5] 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.

[6] N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

[7] N. Hansen, S. P. N. Niederberger, L. Guzzella, and P. Koumoutsakos. A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *IEEE Transactions on Evolutionary Computation*, 200? Accepted for publication.

[8] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[9] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

[10] R. Salomon. Evolutionary algorithms and gradient search: similarities and differences. *Evolutionary Computation, IEEE Transactions on*, 2(2):45–55, 1998.

[11] R. Salomon and J. L. van Hemmen. Accelerating backpropagation through dynamic self-adaptation. *Neural Networks*, 9(4):589–601, 1996.

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