



On the parallel speed-up of Estimation of Multivariate Normal Algorithm and Evolution Strategies

Fabien Teytaud, Olivier Teytaud

► To cite this version:

Fabien Teytaud, Olivier Teytaud. On the parallel speed-up of Estimation of Multivariate Normal Algorithm and Evolution Strategies. EvoNum (evostar workshop), 2009, Tuebingen, Germany. inria-00369781

HAL Id: inria-00369781

<https://inria.hal.science/inria-00369781>

Submitted on 21 Mar 2009

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.

On the parallel speed-up of Estimation of Multivariate Normal Algorithm and Evolution Strategies

F. Teytaud and O. Teytaud

TAO (Inria), LRI, UMR 8623(CNRS - Univ. Paris-Sud), bat 490 Univ. Paris-Sud
91405 Orsay, France, fteytaud@lri.fr

Abstract. Motivated by parallel optimization, we experiment EDA-like adaptation-rules in the case of λ large. The rule we use, essentially based on estimation of multivariate normal algorithm, is (i) compliant with all families of distributions for which a density estimation algorithm exists (ii) simple (iii) parameter-free (iv) better than current rules in this framework of λ large. The speed-up as a function of λ is consistent with theoretical bounds.

1 Introduction

Evolution Strategies (ES [16]) are a robust optimization tool, known for its robustness (in particular in front of local minima) and simplicity. It is also known as suitable for parallel optimization, because it is population-based. However, it has been pointed out recently in [4] that usual step-size adaptation rules were far from being efficient for λ large, e.g. $\lambda = 4N^2$ where λ is the population size and N is the dimension of the search space.

The case of $\lambda = 4N^2$ as in [4] or $\lambda \gg 4N^2$ is not purely theoretical. In our recent applications, we usually optimized on a cluster of 368 cores, and we recently organized an optimization on several clusters of a few hundreds cores on each cluster. With $\lambda = 2000$ cores, $N = 22$ satisfies $\lambda = 4N^2$. Moreover, in many cases we have to launch several jobs simultaneously (more than the number of cores) in order to save up scheduling time, leading to λ much larger than the number of cores, and in robust optimization $N \leq 50$ is far from being trivial.

In this paper, we: (i) describe the main step-size adaptation rules in the literature (section 2); (ii) experiment step-size adaptation rules for various values of λ (section 3); (iii) discuss the results in section 4.

In all the paper, we assume that the reader is familiar with standard notations around ES (see e.g. [16, 2] for more information) and we consider $(\mu/\mu, \lambda)$ -algorithms, i.e.: (i) at each iteration of the algorithm, λ points are generated according to some (usually but not necessarily) Gaussian distribution; (ii) the fitness values of these λ points are computed; (iii) then, the μ best points according to the fitness function are selected; averages are then w.r.t this subsample of the μ best points. All averages here are unweighted averages, but methods used in this paper can be applied in weighted cases. $N(0, Id)$ and related notations

(e.g. $N_i(0, Id)$) denote standard multivariate Gaussian random variables with identity covariance matrix.

2 Methods

A central issue in ES is the adaptation of the distribution (step-size and beyond). The one-fifth rule has been successfully introduced in [16] for this, and several other rules have been proposed later in the literature; some main rules are detailed in the rest of this section.

In this paper, we are considering minimization problems.

2.1 Cumulative Step-size Adaptation (CSA)

Cumulative step-size adaptation has been proposed in [9]; essentially, this method compares the path followed by the algorithm to the path followed under random selection: if the followed path is larger, then the step size should increase. The detailed algorithm is presented in Algorithm 1.

Algorithm 1 Cumulative step-size adaptation.

```

Initialize  $\sigma \in \mathbb{R}, y \in \mathbb{R}^N$ .
while halting criterion not fulfilled do
  for  $i = 1..\lambda$  do
     $s_i = N_i(0, Id)$ 
     $y_i = y + \sigma s_i$ 
     $f_i = f(y_i)$ 
  end for
  Sort the individuals by increasing fitness;  $f_{(1)} < f_{(2)} < \dots < f_{(\lambda)}$ .
   $s^{avg} = \frac{1}{\mu} \sum_{i=1}^{\mu} s_{(i)}$ 
   $y = y + \sigma s^{avg}$ 
   $p_\sigma = (1 - c)p_\sigma + \sqrt{\mu c(2 - c)}s^{avg}$ 
   $\sigma = \sigma \exp[\frac{\|p_\sigma\| - \overline{\chi_N}}{D\overline{\chi_N}}]$ 
end while

```

where $\overline{\chi_N}$ is $\sqrt{N} \times (1 - \frac{1}{4.0 \times N} + \frac{1}{21.0 \times N^2})$. Following [8], we choose $c = \frac{1}{\sqrt{N}}$ and $D = \sqrt{N}$. A main weakness of this formula is its moderate efficiency, for λ large, as pointed out in [4]. [4] therefore proposes mutative self-adaptation in order to improve the convergence rate as a function of λ ; this mutative self-adaptation is presented below.

2.2 Mutative Self-Adaptation (SA)

Mutative self-adaptation has been proposed in [16] and [19], and extended in [4] for improving the convergence rate for λ large. The algorithm is simple and provides the state of the art results for λ large; it is presented in Algorithm 2.

Algorithm 2 Mutative self-adaptation. τ is equal to $1/\sqrt{N}$; other variants have been tested (e.g. $1/\sqrt{2N}$ which is sometimes found in papers) without improvement in our case of λ large.

```

Initialize  $\sigma^{avg} \in \mathbb{R}, y \in \mathbb{R}^N$ .
while Halting criterion not fulfilled do
  for  $i = 1..\lambda$  do
     $\sigma_i = \sigma^{avg} e^{\tau N_i(0,1)}$ 
     $z_i = \sigma_i N_i(0, Id)$ 
     $y_i = y + z_i$ 
     $f_i = f(y_i)$ 
  end for
  Sort the individuals by increasing fitness;  $f_{(1)} < f_{(2)} < \dots < f_{(\lambda)}$ .
   $z^{avg} = \frac{1}{\mu} \sum_{i=1}^{\mu} z_{(i)}$ 
   $\sigma^{avg} = \frac{1}{\mu} \sum_{i=1}^{\mu} \sigma_{(i)}$ 
   $y = y + z^{avg}$ 
end while

```

2.3 Statistical Step-size Adaptation (SSA)

We here propose simple step-size adaptation rules, inspired by Estimation of Distribution Algorithms (EDA), e.g. UMDA [14], Compact Genetic Algorithm [10], Population-Based Incremental Learning [1], Relative Entropy [13], Cross-Entropy [5] and Estimation of Multivariate Normal Algorithms (EMNA) [11] (our main inspiration), which combine (i) the current distribution (possibly), (ii) statistical properties of selected points, into a new distribution. We show in this paper that forgetting the old estimate and only using the new points is a good idea in the case of λ large; in particular, premature convergence as pointed out in [20, 7, 12, 15] does not occur if $\lambda \gg 1$ points are distributed on the search space with non-degenerated variance, and troubles around variance estimates for small sample size as in [6] are by definition not relevant for us. Its advantages are as follows for λ large: (i) it's very simple and parameter free; the reduced number of parameters is an advantage of mutative self adaptation in front of cumulative step-size adaptation, but we show here that yet fewer parameters (0!) is possible and leads to better results, thanks to EMNA-like algorithms; (ii) we use it here in the case of a Gaussian, but it could easily be used also for any density estimation technique e.g. sums of Gaussians (i.e. no problem with multimodal distributions); (iii) it could easily be used for discrete spaces also. The algorithm is presented in Algorithm 3 for estimating only one step-size, but a straightforward extension is one step-size per axis or even a full covariance matrix (see section 3.3).

3 Experimental results

As we here focus on step-size adaptation-rules, we here do not consider complete Covariance-Matrix Adaptation (CMA) but only diagonal covariance matrix

Algorithm 3 SSA. This is close to EMNA; we divide by $\mu \times N$ in the adaptation rule (eq. defining σ) because the sum is over $\mu \times N$ deviations (one per selected individual and per axis), and we add a trick against premature convergence. The constant K is here ∞ as this work is not devoted to premature convergence; however, we verified that the same convergence rates as those presented in this paper can be recovered with this more robust version.

```

Initialize  $\sigma \in \mathbb{R}, y \in \mathbb{R}^N$ .
while Halting criterion not fulfilled do
  for  $l = 1..\lambda$  do
     $z_l = \sigma N_l(0, Id)$ 
     $y_l = y + z_l$ 
     $f_l = f(y_l)$ 
  end for
  Sort the individuals by increasing fitness;  $f_{(1)} < f_{(2)} < \dots < f_{(\lambda)}$ .
   $z^{avg} = \frac{1}{\mu} \sum_{i=1}^{\mu} z_{(i)}$ 
  if  $\|z^{avg}\| < K\sigma$  then
    
$$\sigma = \sqrt{\frac{\sum_{i=1}^{\mu} \|z_{(i)} - z^{avg}\|^2}{\mu \times N}}$$

  else
     $\sigma = 2\sigma$   /** this avoids premature convergence in case of  $\sigma$  poorly chosen**/
  end if
   $y = y + z^{avg}$ 
end while

```

adaptation, but [17] has shown that this diagonal framework is indeed better in many (even non separable) cases, and we point out that all methods discussed in section 2 have anyway straightforward extensions in the framework of complete covariance matrix or covariance matrix by block. We first confirm results from [4] (superiority of mutative self-adaptation over cumulative step-size adaptation for λ large), in section 3.1; we then validate the statistical step-size adaptation rule in section 3.2. Table 1 presents the objective functions considered in this paper.

3.1 Validating SA for λ large

We here confirm results in [4], namely the moderate efficiency of CSA for λ large (at least under the various parameterizations we tested). Following [4], Figure 1 presents the numbers of iterations before a fixed halting criterion ($f_{stop} = 10^{-10}$).

3.2 Validating the statistical step-size adaptation

We now present the main result of this paper, i.e. the efficiency of the statistical step-size adaptation rule. Results are presented in Fig. 2, 3, 4 for $\mu = \lambda/2$,

Name	Objective function
Sphere	$f(y) = \sum_{i=1}^N y_i^2$
Schwefel	$f(y) = \sum_{i=1}^N (\sum_{j=1}^i y_j)^2$
Cigar	$f(y) = y_1^2 + 10000 \times \sum_{i=2}^N y_i^2$

Table 1. Objective functions considered in this paper.

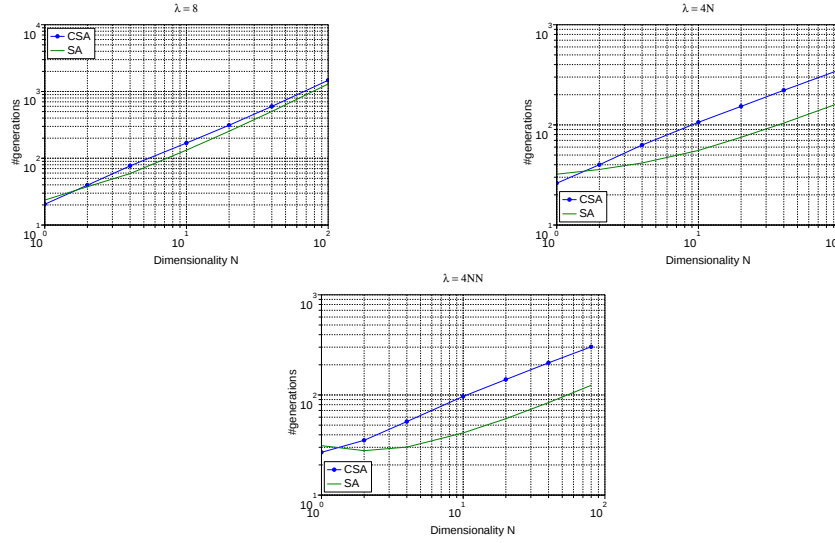


Fig. 1. Performance of CSA and SA on the sphere function (number of iterations before $f(x) < 10^{-10}$) depending on the dimensionality. We confirm here the superiority of SA on cumulative step-size adaptation, at least for our version of SA and CSA. In these experiments, $\mu = \frac{1}{4}\lambda$.

$\mu = \lambda/4$, $\mu = 1$ respectively. Presented curves are $N \times \log(\|x - x^*\|)/n$ as a function of λ , where: N is the dimensionality; x is the best point in the last λ offspring; x^* is the optimum ($x^* = 0$ for our test functions); n is the number of iterations before the halting criterion is met. Each run is performed until $f(x) < 10e^{-50}$. For the sake of statistical significance each point is the average of 300 independent runs.

3.3 Anisotropic case

The SSA algorithm can be adapted to the anisotropic case. We here consider the separable version, i.e. a diagonal covariance matrix for the Gaussian; this form of covariance matrix is intermediate between the full matrix and the matrix proportional to identity. It has been pointed out in [17] that this separable

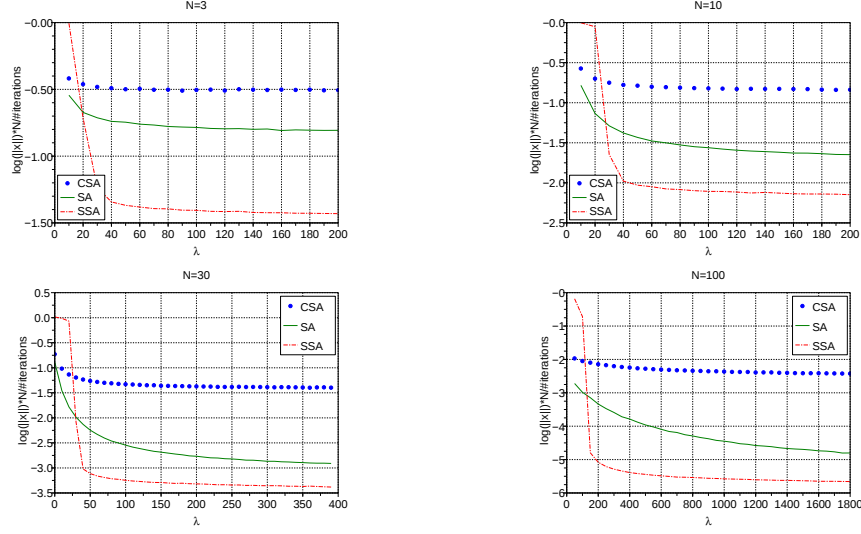


Fig. 2. Log of distance to the optimum when the halting criterion (see text) is met, normalized by the dimension and the number of iterations. Results for $\mu = \lambda/2$, on the sphere function. In all cases, SSA is the best rule for λ large.

version is in many cases faster than the complete covariance matrix. We have in this case to determine one step-size per axis; see Algorithm 4.

Figure 5 presents the results of Algorithm 4. We can see that: (i) On the Schwefel function, the results are the same as in the case of the sphere function. The isotropic algorithms have, in this framework (non presented results), a result close to 0 (i.e. much worse). Therefore, the anisotropic step-size adaptation works for this moderately ill-conditioned function. (ii) On the Cigar function, the results are not exactly those of the sphere function, but almost; even on this much more ill-conditioned function, the anisotropic SSA works.

4 Discussion

First, we confirm the superiority of mutative self-adaptation over cumulative step-length adaptation, as already shown in [4], in the case of λ large. However, possibly, tuning CSA might lead to improved results for high values of λ ; this point, beyond the scope of this paper, is not further analyzed here. Second, we experiment a simple statistical adaptation rule adapted from EMNA. This rule for step-size adaptation is (i) very simple (the most simple of all rules in section 2) (ii) fully portable to the full-covariance case (as well as the SA approach used in [4] for covariance matrix adaptation) (iii) computationally cheap (iv) highly intuitive (v) parameter-free (except K if we include the rule against premature convergence, which is only required if choosing σ sufficiently large is hard) (vi) efficient whenever λ is not known in advance as λ has no

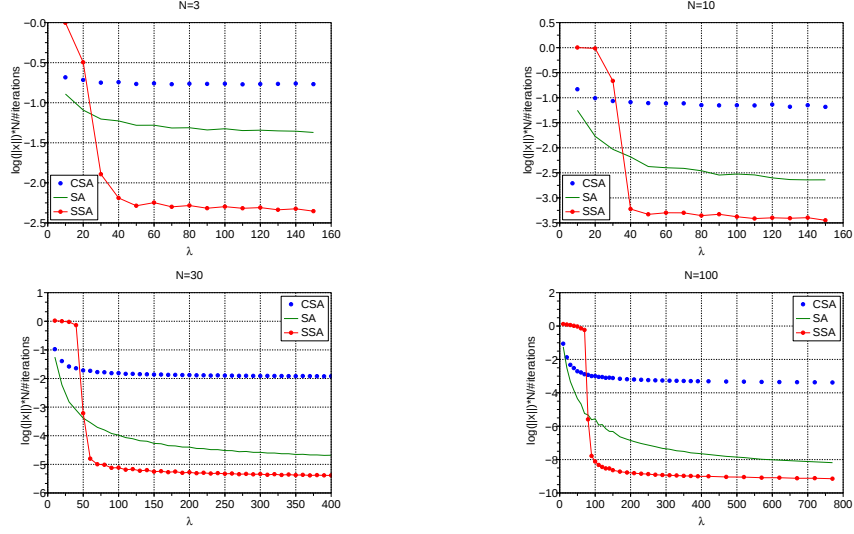


Fig. 3. Results for $\mu = \lambda/4$ on the sphere function. All methods are better than for $\mu = \lambda/2$. In all cases, SSA is the best rule for λ large.

impact on the adaptation (important for fault-tolerance). It provides a speed-up of 100% (over mutative self-adaptation, which is already much better than CSA) on the sphere function for $N = 3$, decreasing to 33% for $N = 10$ and 25% for $N = 100$ (in the case $\lambda = 150$). The usual adaptation-rules use a combination of an old covariance matrix and a new estimate based on the current population - essentially, in SSA, we only keep the second term as λ large simplifies the problem. We point out that we only experimented the Gaussian case so that we can compare our results to standard rules, but our algorithm has the advantage that it can be adapted to *all* distributions for which the parameters can be estimated from a sample. Sums of Gaussians are a natural candidate for further work. Third, we show that the anisotropic version works in the sense that the convergence rate on the sphere was recovered with the Schwefel and the Cigar function.

Some by-products of our results are now pointed out, as a comparison with theoretical results in [21]: (i) The higher the dimensionality, the better the speed-up for $(\mu/\mu, \lambda)$ -algorithms; [3] has shown the linear speed-up as a function of λ as long as $\lambda \ll N$, and [21] has precised formally that the speed-up is linear until λ of the same order as the dimension. Roughly, a number of processors linear as a function of the dimension is efficient. This is visible on our experimental results. (ii) Also, $(1, \lambda)$ algorithms (case $\mu = 1$) have a less-than-linear (logarithmic) speed-up as a function of λ . This is visible in our experimental results. This is also consistent with [3] and [21]. (iii) We have proposed a very simple rule and got state-of-the-art results. This suggests that there is much to win by a refined work on the important case of λ large.

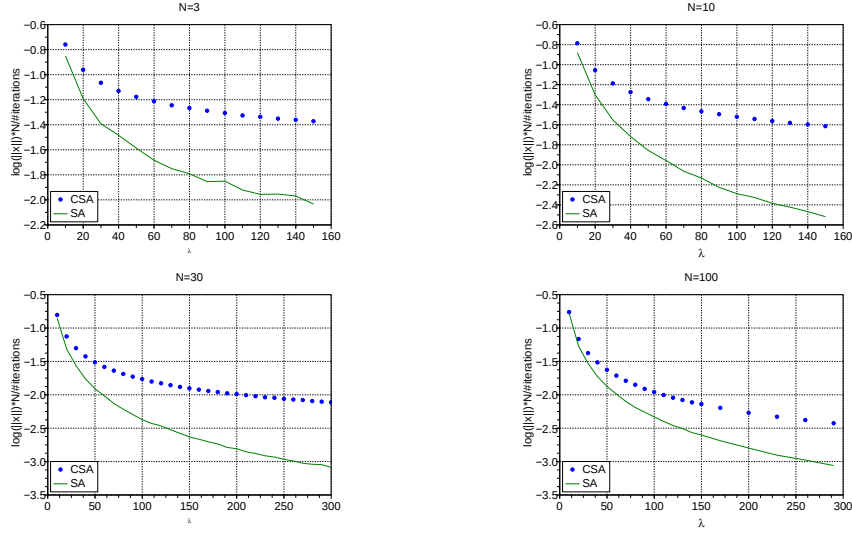


Fig. 4. Results for $\mu = 1$ on the sphere function. SSA is not presented as it does not make sense for $\mu = 2$. As shown by this figure (compared to Figs 3 and 2), $\mu = 1$ is quite weak for large dimension and the absence of SSA version for that case is therefore not a trouble.

Acknowledgements We are grateful to H.-G. Beyer for useful email answers, and to A. Auger, N. Hansen, R. Ros for fruitful discussions. We also thank the MoGo-people (G. Chaslot, J.-B. Hoock, A. Rimmel) for interesting discussions around parallel applications of ES.

References

1. S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning,. Technical Report CMU-CS-94-163, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1994.
2. H.-G. Beyer. *The Theory of Evolution Strategies*. Natural Computing Series. Springer, Heidelberg, 2001.
3. H.-G. Beyer. *The Theory of Evolutions Strategies*. Springer, Heidelberg, 2001.
4. H.-G. Beyer and B. Sendhoff. Covariance matrix adaptation revisited - the CMSA evolution strategy. In G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, editors, *Proceedings of PPSN*, pages 123–132, 2008.
5. Y. Cai, X. Sun, H. Xu, and P. Jia. Cross entropy and adaptive variance scaling in continuous eda. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 609–616, New York, NY, USA, 2007. ACM.
6. W. Donga and X. Yao. Unified eigen analysis on multivariate gaussian based estimation of distribution algorithms. *Information Sciences*, 178(15):3000–3023, 2008.

Algorithm 4 Anisotropic statistical step-size adaptation. We divide the average squared deviation by μ in the step-size adaptation rule (equation defining σ), instead of $\mu \times N$ in the anisotropic case, because now the step-size is averaged over μ squared deviations only (one per selected individual). The subscript j refers to the axis.

```

Initialize  $\sigma \in \mathbb{R}^N, y \in \mathbb{R}^N$ .
while Halting criterion not fulfilled do
  for  $l = 1..N$  do
    for  $i \in \{1, \dots, N\}$  do
       $z_{l,i} = \sigma_i N_{l,i}(0, 1)$ 
    end for
     $y_l = y + z_l$ 
     $f_l = f(y_l)$ 
  end for
  Sort the individuals by increasing fitness;  $f_{(1)} < f_{(2)} < \dots < f_{(N)}$ .
   $z^{avg} = \frac{1}{\mu} \sum_{i=1}^{\mu} z_{(i)}$ 
  for  $j \in \{1, \dots, N\}$  do
     $\sigma_j = \sqrt{\frac{\sum_{i=1}^{\mu} \|z_{(i),j} - z_j^{avg}\|^2}{\mu}}$ 
  end for
   $y = y + z^{avg}$ 
end while

```

7. J. Grahl, P. A. Bosman, and F. Rothlauf. The correlation-triggered adaptive variance scaling idea. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 397–404, New York, NY, USA, 2006. ACM.
8. N. Hansen. *Verallgemeinerte individuelle Schrittweitenregelung in der Evolutionsstrategie. Eine Untersuchung zur entstochastisierten, koordinatensystem-unabhängigen Adaptation der Mutationsverteilung*. Mensch und Buch Verlag, Berlin, 1998. ISBN 3-933346-29-0.
9. N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
10. G. R. Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE Trans. on Evolutionary Computation*, 3(4):287, November 1999.
11. P. Larranaga and J. A. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.
12. J. Liu and H.-F. Teng. Model learning and variance control in continuous edas using pca. In *ICICIC '08: Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control*, page 555, Washington, DC, USA, 2008. IEEE Computer Society.
13. H. Mühlenbein and R. Höns. The estimation of distributions and the minimum relative entropy principle. *Evolutionary Computation*, 13(1):1–27, 2005.
14. H. Mühlenbein and T. Mahnig. Evolutionary computation and Wright’s equation. *Theoretical Computer Science*, 287(1):145–165, 2002.

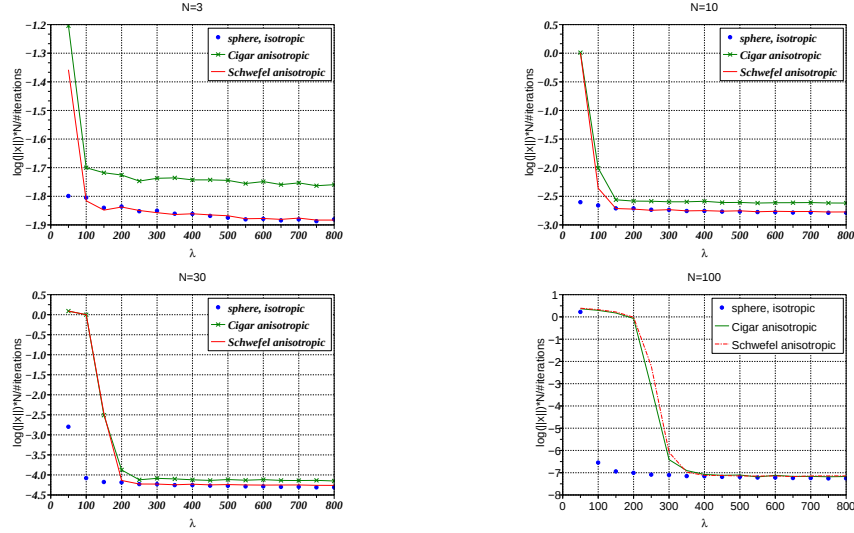


Fig. 5. We here present the results of the anisotropic version of the SSA algorithm (Algorithm 4). It has on ill-conditioned functions nearly the same performance as the isotropic version on the sphere, in particular in high dimension.

15. P. Posík. Preventing premature convergence in a simple eda via global step size setting. In Rudolph et al. [18], pages 549–558.
16. I. Rechenberg. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution*. Fromman-Holzboog Verlag, Stuttgart, 1973.
17. R. Ros and N. Hansen. A simple modification in cma-es achieving linear time and space complexity. In Rudolph et al. [18], pages 296–305.
18. G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, editors. *Parallel Problem Solving from Nature - PPSN X, 10th International Conference Dortmund, Germany, September 13-17, 2008, Proceedings*, volume 5199 of *Lecture Notes in Computer Science*. Springer, 2008.
19. H.-P. Schwefel. Adaptive Mechanismen in der biologischen Evolution und ihr Einfluss auf die Evolutionsgeschwindigkeit. Interner Bericht der Arbeitsgruppe Bionik und Evolutionstechnik am Institut für Mess- und Regelungstechnik Re 215/3, Technische Universität Berlin, Juli 1974.
20. J. L. Shapiro. Drift and scaling in estimation of distribution algorithms. *Evolutionary Computation*, 13(1), 2005.
21. O. Teytaud and H. Fournier. Lower bounds for evolution strategies using vc-dimension. In Rudolph et al. [18], pages 102–111.