Convergence Properties of Evolution Strategies with the Derandomized Covariance Matrix Adaptation: The $(\mu/\mu_{\mathrm{I}}, \lambda)$-CMA-ES

Nikolaus Hansen & Andreas Ostermeier

Fachgebiet für Bionik und Evolutionstechnik

Technische Universität Berlin

Ackerstr. 71–76, 13355 Berlin, Germany

E-mail: {hansen,ostermeier}@fb10.tu-berlin.de

In: *EUFIT'97, 5th European Congress on Intelligent Techniques and Soft Computing,*

Proceedings: 650–654

ftp: ftp-bionik.fb10.tu-berlin.de/pub/papers/Bionik/CMAES2.ps.Z

Convergence Properties of Evolution Strategies with the Derandomized Covariance Matrix Adaptation: The $(\mu/\mu_{\mathrm{I}}, \lambda)$-CMA-ES

Nikolaus Hansen and Andreas Ostermeier
Technische Universität Berlin, Sekr. ACK 1
Fachgebiet für Bionik und Evolutionstechnik
Ackerstr. 71–76, 13355 Berlin, Germany
phone:+30 / 314 72666, fax:+30 / 314 72658
e-mail: {hansen,ostermeier}@fb10.tu-berlin.de

Abstract: The intermediate (center of mass) recombination of object parameters is introduced in the evolution strategy with derandomized covariance matrix adaptation (CMA-ES). On various (unimodal) real space fitness functions convergence properties and robustness against distorted selection are tested for different parent numbers. Introducing $(\mu/\mu_{\mathrm{I}}, \lambda)$-selection significantly improves robustness and has comparatively minor influence on the convergence speed of the CMA-ES.

I. Introduction

In evolutionary computation usually the aspect of *global* search is emphasized. In contrast, to approach a (global) optimum quickly, after its basin of attraction is found, convergence velocity of an evolutionary search algorithm is important. In our opinion the importance of this aspect is, at least in continuous parameter optimization problems, underestimated; in practical applications the optimum is not approached sufficiently—'premature convergence' can be observed. In evolution strategies (ESs) this problem is adressed by the adaptation of the mutation distribution, e.g. global or individual step size control [5; 6]. A generalized approach to individual step size control adapts the coordinate system, in which the step size control takes place, as well as the step sizes. This allows to generate any normal mutation distribution with zero mean.

Such a scheme was realized the first time in [6], putting $n(n-1)$ angles for rotating the coordinate system under the control of mutation and selection. It is disappointing to notice, that this scheme highly depends on orientation and permutation(!) of the coordinate axes [3; 4]. The rotations, performed *in the canonical planes*, are the reason for that dependency. Consequently, the scheme lacks the ability to adapt for topologies of quadratic functions which are badly scaled and not axis parallel orientated (e.g. functions 4–6 in Table 1, Sect. III), resulting in a very poor performance [4].

A different approach to a generalized individual step size control is the (derandomized) covariance matrix adaptation (CMA), introduced in [2]. The CMA is independent of the given coordinate system, reliably adapts topologies of arbitrary quadratic functions and significantly improves convergence rates especially on non-separable and/or badly scaled fitness functions.

In this paper we combine intermediate (center of mass) recombination of the object parameters with the covariance matrix adaptation resulting in the $(\mu/\mu_{\mathrm{I}}, \lambda)$-CMA-ES. Advantages of intermediate μ/μ_{I} recombination can be twofold. On the one side theoretical results foresee a speed up of progress, assuming sufficiently large problem dimension and optimal step length [5; 1]. On the other side robustness against selection error should increase remarkably.

In Sect. II we introduce the $(\mu/\mu_{\mathrm{I}}, \lambda)$-CMA-ES. Section III discusses test functions. In Sect. IV simulation results are shown and Sect. V provides concluding remarks.

II. Algorithm: The $(\mu/\mu_{\mathrm{I}}, \lambda)$-CMA-ES

Every new object parameter vector $\mathbf{x}_k^{(g+1)}, k = 1 \ldots \lambda$, of generation $g+1$ is generated by adding a realisation of a $\mathcal{N}\left(\mathbf{0}, {\delta^{(g)}}^2 \mathbf{C}^{(g)}\right)$ distributed random vector. The vector is generated by linear transformation of $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathbf{I}$ is the identity matrix. For $k = 1 \ldots \lambda$ we yield

$$\mathbf{x}_k^{(g+1)} = \langle \mathbf{x} \rangle_\mu^{(g)} + \delta^{(g)} \mathbf{B}^{(g)} \mathbf{D}^{(g)} \mathbf{z}_k \tag{1}$$

where

$\mathbf{x}_k^{(g+1)} \in \mathbb{R}^n$. Object variable vector of k^{th} individual at generation $g+1$.

$\langle \mathbf{x} \rangle_\mu^{(g)} = \frac{1}{\mu} \sum_{j \in I_{\text{sel}}}^{(g)} \mathbf{x}_j^{(g)}$. Center of mass of the μ selected (best) individuals of generation g. $I_{\text{sel}}^{(g)}$ is the set of indices of the selected individuals at generation g, $|I_{\text{sel}}| = \mu$.

$\delta^{(g)}$ Step size.

$\mathbf{B}^{(g)}$ Orthogonal $n \times n$-matrix, which linearly transforms $\mathbf{D}^{(g)}\mathbf{z}$. Columns of $\mathbf{B}^{(g)}$ are eigenvectors of the covariance matrix $\mathbf{C}^{(g)}$ (see also below). For any two columns $\mathbf{b}_i$ and $\mathbf{b}_j$, $i \neq j$, of $\mathbf{B}$ holds $\|\mathbf{b}_i\| = 1$ and $\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0$ and therefore $\mathbf{B}^{-1} = \mathbf{B}^{\mathrm{T}}$.

$\mathbf{D}^{(g)}$ Diagonal $n \times n$-matrix. The diagonal element $d_{ii}^{(g)}$ is the square root of an eigenvalue of the covariance matrix $\mathbf{C}^{(g)}$ (see also below). The corresponding eigenvector is the i^{th} column of $\mathbf{B}^{(g)}$. That is, for any column $\mathbf{b}_i^{(g)}$ of $\mathbf{B}^{(g)}$ holds $\mathbf{C}^{(g)}\mathbf{b}_i^{(g)} = {d_{ii}^{(g)}}^2 \mathbf{b}_i^{(g)}$.

$\mathbf{z}_k \in \mathbb{R}^n$. $k = 1 \ldots \lambda$ realizations of a $\mathcal{N}(\mathbf{0}, \mathbf{I})$ distributed random vector, i.e. components of $\mathbf{z}$ are independent identically $(0,1)$-normally distributed.

$\mathbf{D}$ scales the axes of the distribution; isodensity lines of $\mathbf{Dz}$ are coordinate axes parallel (hyper-)ellipsoids. $\mathbf{B}$ determines the new orientation of this ellipsoid. The covariance matrix $\mathbf{C}$ determines $\mathbf{B}$ and $\mathbf{D}$, and is adapted by means of a so called evolution path [2], denoted by $\mathbf{s}$.

$$\mathbf{s}^{(g+1)} = (1-c) \cdot \mathbf{s}^{(g)} + c_{\text{u}} \cdot \frac{\sqrt{\mu}}{\delta^{(g)}} \left(\langle \mathbf{x} \rangle_\mu^{(g+1)} - \langle \mathbf{x} \rangle_\mu^{(g)} \right) \tag{2}$$

$$\mathbf{C}^{(g+1)} = (1 - c_{\text{cov}}) \cdot \mathbf{C}^{(g)} + c_{\text{cov}} \cdot \mathbf{s}^{(g+1)} \left(\mathbf{s}^{(g+1)} \right)^{\mathrm{T}} \tag{3}$$

where

$\mathbf{s} \in \mathbb{R}^n$. Sum of weighted center of mass differences. $\mathbf{s}$ represents the evolution path of the strategy.

$c \in]0;1]$. $1/c$ corresponds to the accumulation time for $\mathbf{s}$. For $c = 1$, $\mathbf{s}^{(g+1)}$ only depends on object parameter vectors of generation g and $g+1$.

$c_{\text{u}} = \sqrt{c \cdot (2-c)}$ normalizes the variance of $\mathbf{s}$ because $1^2 = (1-c)^2 + c_{\text{u}}^2$.

$\mathbf{C}^{(g)}$ Symmetric $n \times n$-matrix, which is the covariance matrix of the normally distributed random vector $\mathbf{B}^{(g)}\mathbf{D}^{(g)}\mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. $\mathbf{C}^{(g)}$ determines $\mathbf{B}^{(g)}$ and $\mathbf{D}^{(g)}$ and $\mathbf{C}^{(g)} = \mathbf{B}^{(g)}\mathbf{D}^{(g)} \left(\mathbf{B}^{(g)}\mathbf{D}^{(g)} \right)^{\mathrm{T}}$.

$c_{\text{cov}} \in [0;1[$. $1/c_{\text{cov}}$ corresponds to the averaging time for the covariance matrix.

Notice, that independent of the size of μ, only one covariance matrix is used.

The step size δ is adapted separately, because changes of overall variance should be done on a much shorter time scale than the adaptation of the covariance matrix. For step size adaptation $\langle \mathbf{x} \rangle_\mu^{(g+1)} - \langle \mathbf{x} \rangle_\mu^{(g)}$ is transformed to reverse the scaling by $\mathbf{D}$, done in (1). This allows to calculate the expected length of $\mathbf{s}_\delta$.

$$\mathbf{s}_\delta^{(g+1)} = (1-c) \cdot \mathbf{s}_\delta^{(g)} + c_{\text{u}} \cdot \mathbf{B}^{(g)} \left(\mathbf{D}^{(g)} \right)^{-1} \left(\mathbf{B}^{(g)} \right)^{-1} \frac{\sqrt{\mu}}{\delta^{(g)}} \left(\langle \mathbf{x} \rangle_\mu^{(g+1)} - \langle \mathbf{x} \rangle_\mu^{(g)} \right) \tag{4}$$

$$\delta^{(g+1)} = \delta^{(g)} \cdot \exp\left(D \, \frac{\|\mathbf{s}_\delta^{(g+1)}\| - \widehat{\chi}_n}{\widehat{\chi}_n} \right) \tag{5}$$

where

$\mathbf{s}_\delta \in \mathbb{R}^n$ represents an evolution path, which is not scaled by $\mathbf{D}$.

$\mathbf{D}^{-1}$ can easily be calculated by inverting the diagonal elements of $\mathbf{D}$ individually.

$\mathbf{B}^{-1} = \mathbf{B}^{\mathrm{T}}$.

$D \in]0;1]$. Parameter for damping the step size variation.

$\widehat{\chi}_n = \sqrt{n}\,(1 - \frac{1}{4\,n} + \frac{1}{21\,n^2})$ estimates the expected length of $\mathbf{s}_\delta$ under random selection, which is then $\mathcal{N}(\mathbf{0}, \mathbf{I})$ distributed.

Notice, that (2) and (4) are looking very similar.

The following **parameter setting** is chosen: $c = 1/\sqrt{n}$, $c_{\text{cov}} = 2/(n^2+n)$, $D = 1/\sqrt{n}$ and $\lambda = 10$, while start values are $\mathbf{s}^{(0)} = \mathbf{0}$, $\mathbf{s}_\delta^{(0)} = \mathbf{0}$ and $\mathbf{C}^{(0)} = \mathbf{I}$. $\delta^{(0)}$ and $\langle \mathbf{x} \rangle_\mu^{(0)}$ are chosen with respect to the fitness function (cf. Table 1). The comparatively small λ is preferable because adaptation time mostly depends on the generation number and improvement per generation usually scales sub-linear with increasing λ (fixed $n \ll \infty$).

Table 1. Test Functions (to be minimized)

Name	Formula	$\delta^{(0)}$	$\langle \mathbf{x} \rangle_\mu^{(0)}$	f^{stop}
1) Sphere	$f_{\mathrm{sphere}}(\mathbf{x}) = \sum_{i=1}^{n} (x_i - 1)^2$	1	$\mathbf{0}$	10^{-10}
2) Schwefel's Problem	$f_{\mathrm{schwef}}(\mathbf{y}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} y_i \right)^2$	1	$\sum_{i=1}^{n} \mathbf{o}_i$	10^{-10}
3) Rosenbrock's Function	$f_{\mathrm{rosen}}(\mathbf{y}) = \sum_{i=1}^{n-1} \left(100 \left({y_i}^2 - y_{i+1} \right)^2 + (y_i - 1)^2 \right)$	0.1	$\mathbf{0}$	10^{-10}
4) Ellipsoid	$f_{\mathrm{elli}}(\mathbf{y}) = \sum_{i=1}^{n} \left(1000^{\frac{i-1}{n-1}} y_i \right)^2$	1	$\sum_{i=1}^{n} \mathbf{o}_i$	10^{-10}
5) Cigar	$f_{\mathrm{cigar}}(\mathbf{y}) = {y_1}^2 + \sum_{i=2}^{n} (1000\, y_i)^2$	1	$\sum_{i=1}^{n} \mathbf{o}_i$	10^{-10}
6) Tablet	$f_{\mathrm{tablet}}(\mathbf{y}) = (1000\, y_1)^2 + \sum_{i=2}^{n} {y_i}^2$	1	$\sum_{i=1}^{n} \mathbf{o}_i$	10^{-10}
7) Different Powers	$f_{\mathrm{diffpow}}(\mathbf{y}) = \sum_{i=1}^{n} \lvert y_i \rvert^{2+10\frac{i-1}{n-1}}$	0.1	$\sum_{i=1}^{n} \mathbf{o}_i$	10^{-15}
8) Parabolic Ridge	$f_{\mathrm{parab}}(\mathbf{y}) = -y_1 + \sum_{i=2}^{n} {y_i}^2$	1	$\mathbf{0}$	-10^5
9) Sharp Ridge	$f_{\mathrm{sharp}}(\mathbf{y}) = -y_1 + 100 \sqrt{\sum_{i=2}^{n} {y_i}^2}$	1	$\mathbf{0}$	-10^5

To our experience, parameter settings are robust if $c_{\mathrm{cov}} \ll D \leq c$, $c_{\mathrm{cov}} < 3/(n^2+n)$, $\lambda > 8$ and $\mu \leq \lambda/2$. **Difficult problems** and/or distorted selection may require smaller c_{cov} and/or larger λ than stated. For $\mu = 1$ the algorithm is *exactly* the same as in [2] if one chooses $D = (1 - \frac{1}{4\,n} + \frac{1}{21\,n^2})/\sqrt{n}$ and $c_{\mathrm{cov}} = 2/n^2$.

III. Building Simpler Test Functions

According to [7] we want a test function to be nonlinear, non-separable, scalable and resistant to simple hill climbing. In addition, we require an easy to analyze topology even for $n \geq 3$. The last point is often neglected, but is important to reveal what demand on the search algorithm is actually tested by a specific function.

Table 1 gives the test functions used. Only topologies of functions 2, 3 and 7 are not completely transparent. To yield non-separability for functions 4–9 we set $y_i := \langle \mathbf{x}, \mathbf{o}_i \rangle$, where $\mathbf{x}$ is the object parameter vector to be optimized. $\mathbf{o}_1 \ldots \mathbf{o}_n$ is a randomly orientated orthonormal basis, fixed for each run.[1] For the canonical basis, the argument reduces to $y_i = \langle \mathbf{x}, \mathbf{o}_i \rangle = x_i$. For a non-canonical basis only the sphere remains separable. *All results in this paper are independent of the basis actually chosen, i.e. valid for any basis!* The axis scaling between longest and shortest axis on problems 4–6 is 10^3, which is a reasonable supposition of misscaling for a real world problem. Problem 6 (Tablet) can be interpreted as a sphere model with constraints in $\mathbf{o}_1$ direction. To solve the quadratic problems 4–6 should be a minimum demand on a search strategy in $\mathbb{R}^n$. Even thought this test suite demands comparatively simple search strategies (e.g. following a narrow valley or searching in purely quadratic topologies) most ES variants will fail in solving these functions in reasonable time even for small dimensions.

IV. Results and Discussion

The $(\mu/\mu_{\mathrm{I}}, 10)$-CMA-ES is compared to a simple $(\mu/\mu_{\mathrm{I}}, 10)$-ES with isotropic mutation distribution. For the simple ES the CMA-ES algorithm of Sect. II is used setting $c_{\mathrm{cov}} = 0$. This ES still significantly benefits from the efficient global step size control in (4) and (5) then. For simulations with $n > 50$, the update of $\mathbf{B}$ and $\mathbf{D}$ (CMA-ES) were not done at every generation but only at generations $n, 2n, 3n, \ldots$ This reduces the computational effort of the algorithm from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ and may be done, if the computation time for the algorithm exceeds the computation time for the fitness function evaluations, which is unusual in real world problems. Updating the matrices every generation usually yields better results.

Diagrams of **Fig. 1** show the number of function evaluations to reach f^{stop} depending on μ for different dimensions. Shown are mean values and estimated standard deviations from 2 to 30 runs. Where points of graphs for the CMA-ES are missing, f^{stop} was not reached in every run. The mutation distribution then degenerates into a subspace because progress becomes too small and selection information covers the problem topology insufficiently.[2] This only happens for large μ especially in small dimensions.[3] Unsurprisingly, the effect is very similar to the effect seen for distorted selection. Due to limited computational time results for ES with $n > 5$ on functions 4–9 are omitted. To avoid numerical precision problems on f_{sharp}, a lower bound for δ was established at 10^{-10} here.

[1] Each $\mathbf{o}_i$ is equally distributed on the unit (hyper-)sphere surface, dependently drawn so that $\langle \mathbf{o}_i, \mathbf{o}_j \rangle = 0$ if $i \neq j$. Algorithm: FOR $i = 1$ TO n 1) Draw components of $\mathbf{o}_i$ $\mathcal{N}(0,1)$ i.i.d. 2) $\mathbf{o}_i := \mathbf{o}_i - \sum_{j=1}^{i-1} \langle \mathbf{o}_i, \mathbf{o}_j \rangle \, \mathbf{o}_j$ 3) $\mathbf{o}_i := \mathbf{o}_i / \|\mathbf{o}_i\|$ ROF.

[2] For $\mu \to \lambda$ selection information becomes zero.

[3] To prevent this effect, parameter c_{cov} must be chosen smaller, if $\mu > \lambda/2$.

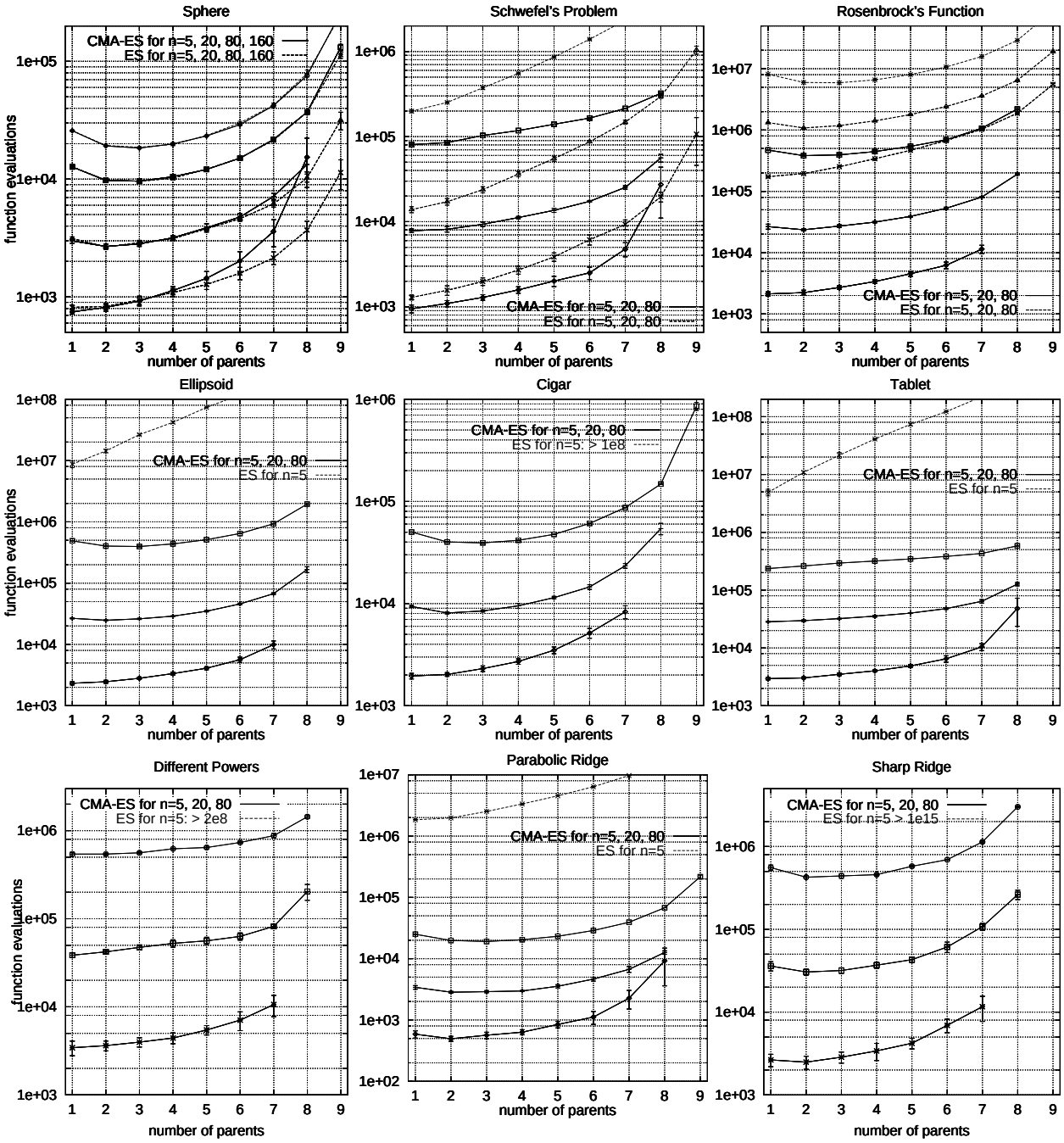

Figure 1. Simulations without selection error. Shown are mean values (2–30 runs) and standard deviations of function evaluations to reach f^{stop} (see Table 1). Missing points of CMA-ES indicate that f^{stop} was not reached reliably.

Convergence speeds of ES and CMA-ES are comparable only on problems 1–3, while on problem 3 (Rosenbrock's function) the CMA-ES is already considerably faster (factor > 10). On problems 4–8 ES takes between 10^3 and 10^5 times longer than CMA-ES to reach f^{stop} for $n = 5$. Problem 9 (sharp ridge) is virtually unsolvable for the simple ES. Speed up factors decrease to 1 for $n \to \infty$, if the convergence time scales better than $\mathcal{O}(n^2)$, the adaptation time for the covariance matrix (cp. results on Rosenbrock's function vs. Schwefel's problem in Fig. 1). In contrast, speed up factors increase (partly drastically), if f^{stop} is chosen to be smaller.

While the covariance matrix adaptation (i.e. parameter c_{cov}) has a drastic effect on the performance, parent number μ, if chosen < 7, is uncritical for CMA-ES performance. Best and worst results for $\mu < 7$ differ by a factor 3 at most. Depending on the problem and problem dimension best results are achieved always for μ between 1 and 3. Notice, that the speed up for $\mu > 1$ (e.g. on the sphere problem) can only be realized

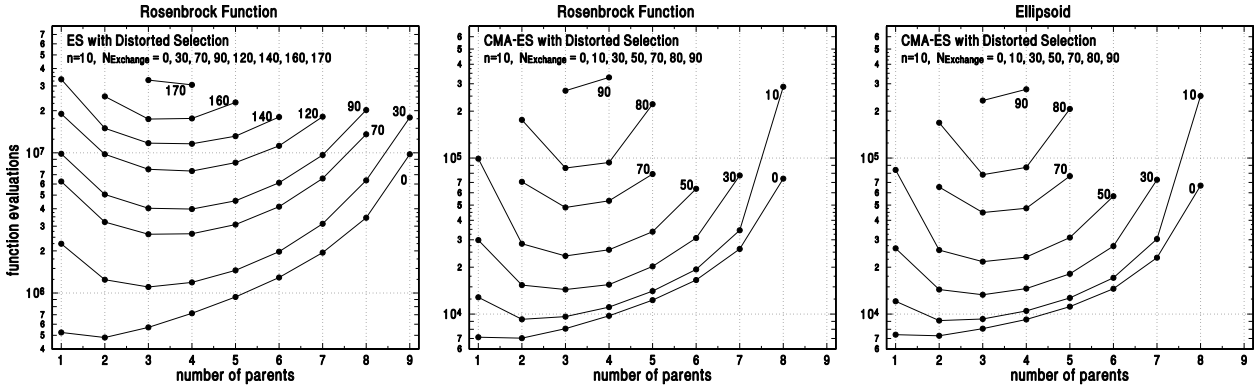

Figure 2. Simulations with distorted selection ($n = 10$). Shown are mean values (5 runs for ES, 50 runs for CMA-ES) of function evaluations to reach f^{stop} (see Table 1). Missing points indicate that f^{stop} was not reached reliably. Notice the different scalings for the axis of ordinate.

because step size control (cf. equations (4) and (5)) adapts nearly optimal step size for *any* given μ. Mutative step size control reduces progress rates significanly by systematically adjusting the step size too small and fails completely for $\mu > \lambda/2$.

Selection distortion is introduced by exchanging adjacent individuals on the ranking scale before choosing the first μ individuals as new parents. Each possible exchange has equal probability. Results are shown in **Fig. 2** for different numbers of exchanges and $n = 10$ on f_{rosen} (ES and CMA-ES) and f_{elli} (CMA-ES only). Again, missing points indicate that f^{stop} was not reached in every run. The figure clearly reveals the advantage of the $(\mu/\mu_{\text{I}}, \lambda)$-selection scheme and shows maximal robustness for $\mu = 3$ in all cases. Behavior on f_{rosen} and f_{elli} is almost identical. CMA-ES is more sensitive to selection error than ES, because the covariance matrix adaptation needs a larger amount of proper selection information.

V. Conclusion

The $(\mu/\mu_{\text{I}}, \lambda)$-CMA-ES, where $\lambda \geq 10$ and $\mu \leq \lambda/2$, efficiently adapts the covariance matrix of the mutation distribution of Evolution Strategies. The adaptation is independent of the chosen coordinate system. Compared to an ES with only global and/or individual step size control, speed up factors between 10 and 10^5 can be expected on non-separable and badly scaled objective functions. To fully exploit the advantages of the method, *at least* $\lambda \cdot n^2$ (more precisely *at least* one to ten times λ/c_{cov}) function evaluations must be done.

The disadvantage of the CMA-ES is the increased sensitivity to the amount of proper selection information. Weak selection, distorted selection or low progress rates can result in a failure of the method caused by a degeneration of the mutation distribution. Under distorted selection choosing $\mu > 1$ can partly make up for this disadvantage. We recommend a $(3/3_{\text{I}}, 12)$-CMA-ES as a fast and, for small λ, most reliable method.

Acknowledgements

This work was supported by the *Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie* under grant 01 IB 404 A.

References

[1] Hans-Georg Beyer. On the asymptotic behavior of multirecombinant evolution strategies. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature—PPSN IV, Proceedings*, pages 122–133, Berlin, 1996. Springer.

[2] Nikolaus Hansen and Andreas 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.

[3] Nikolaus Hansen, Andreas Ostermeier, and Andreas 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*, pages 57–64, San Francisco, 1995. Morgan Kaufmann.

[4] Christian Holzheuer. Analyse der Adaptation von Verteilungsparametern in der Evolutionsstrategie. Master's thesis, Fachgebiet für Bionik und Evolutionstechnik des Fachbereich 6 der Technischen Universität Berlin, September 1996. german language.

[5] Ingo Rechenberg. *Evolutionsstrategie '94*. frommann-holzboog, Stuttgart, 1994.

[6] Hans-Paul Schwefel. *Numerical Optimization of Computer Models*. Wiley, Chichester, 1981.

[7] D. Whitley, K. Mathias, and J. Dzubera. Building better test functions. In L. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 239–246, San Francisco, 1995. Morgan Kaufmann.