Benchmarking sep-CMA-ES on the BBOB-2009 Noisy Testbed

Raymond Ros

Benchmarking sep-CMA-ES on the BBOB-2009 Noisy Testbed

Raymond Ros
Univ. Paris-Sud, LRI
UMR 8623 / INRIA Saclay, projet TAO
F-91405 Orsay, France
raymond.ros@lri.fr

ABSTRACT

A partly time and space linear CMA-ES is benchmarked on the BBOB-2009 noisy function testbed. This algorithm with a multistart strategy with increasing population size solves 10 functions out of 30 in 20-D.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*global optimization, unconstrained optimization*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms

Keywords

Benchmarking, Black-box optimization, Evolutionary computation, Covariance matrix adaptation, Evolution strategy

1. INTRODUCTION

The sep-CMA-ES algorithm introduced in [7] is a variant of the covariance matrix adaptation evolution strategy (CMA-ES) [5] that is linear in time and space. This property combined with a faster learning rate makes sep-CMA-ES appropriate for separable function and larger dimensions. A mixed strategy of using sep-CMA-ES and CMA-ES is proposed here and benchmarked on a noisy function testbed.

2. ALGORITHM PRESENTATION

In its design, the sep-CMA-ES differs from the CMA-ES by two aspects: first, the covariance matrix is constrained to be diagonal at each of its update, second, the learning rate is increased by a factor of $\frac{n+3/2}{3}$, where n is the dimension of the search space [1]. These modifications result in an algorithm that trades model complexity with a time and space scaling that is better than the original CMA-ES. The $(\mu/\mu_W, \lambda)$-sep-CMA-ES has been shown to outperform $(\mu/\mu_W, \lambda)$-CMA-ES on separable functions.

We propose here what would be the best of two worlds: to use sep-CMA-ES for the first few iterations and then switch to CMA-ES. At the time of the switch, all parameters are retained except for the learning rate that is decreased back to its default value. This implies the diagonal covariance matrix acquired using sep-CMA-ES is directly used by CMA-ES. This mixed strategy is therefore expected to be faster than CMA-ES on separable functions. Ongoing work has also shown that for some test functions the first iterations using sep-CMA-ES would not disadvantage the latter use of CMA-ES in any way. In other terms, the cost of initially using sep-CMA-ES would not induce a penalty in the cost of solving the function with CMA-ES afterwards. The author admits some functions could induce such a penalty.

As for the multistart strategy, we use the increasing population size IPOP-CMA-ES [1]. Though this approach has shown its limits [6], independent restart may improve the probability of the algorithm reaching a given target function value.

[1]Please note that the factor for the learning rate is smaller than the one in [7].

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

3. EXPERIMENTAL PROCEDURE

The Matlab implementation of the CMA-ES (version 3.23beta) is used[2]. We use the $(\mu/\mu_W, \lambda)$-IPOP-CMA-ES variant with an initial default population size $\lambda = 4 + \lfloor 3\ln(n) \rfloor$ increasing twice at each restart. Except the learning rate, all other algorithm parameters are set to their default values. The covariance matrix is constrained to be diagonal only for the first $1 + 100n/\sqrt{\lambda}$ iterations of the *first start*. A maximum of 8 independent restarts is conducted. Restarts occur after $100 + 300n\sqrt{n/\lambda}$ iterations or if any of the default stopping criterion is met. The initial stepsize has been set to 2 and the starting point has been chosen uniformly in $[-4, 4]^n$. The maximum number of function evaluations was set to 10^4 times the dimension. No parameter tuning was done, the CrE [3] is computed to zero.

4. RESULTS

Results from experiments according to [3] on the benchmarks functions given in [2, 4] are presented in Figures 1 and 2 and in Tables 1 and 2. From the results of this algorithm, the uniform noise model is the most difficult to deal

[2]Latest version available here:`http://www.lri.fr/~hansen/cmaesintro.html`

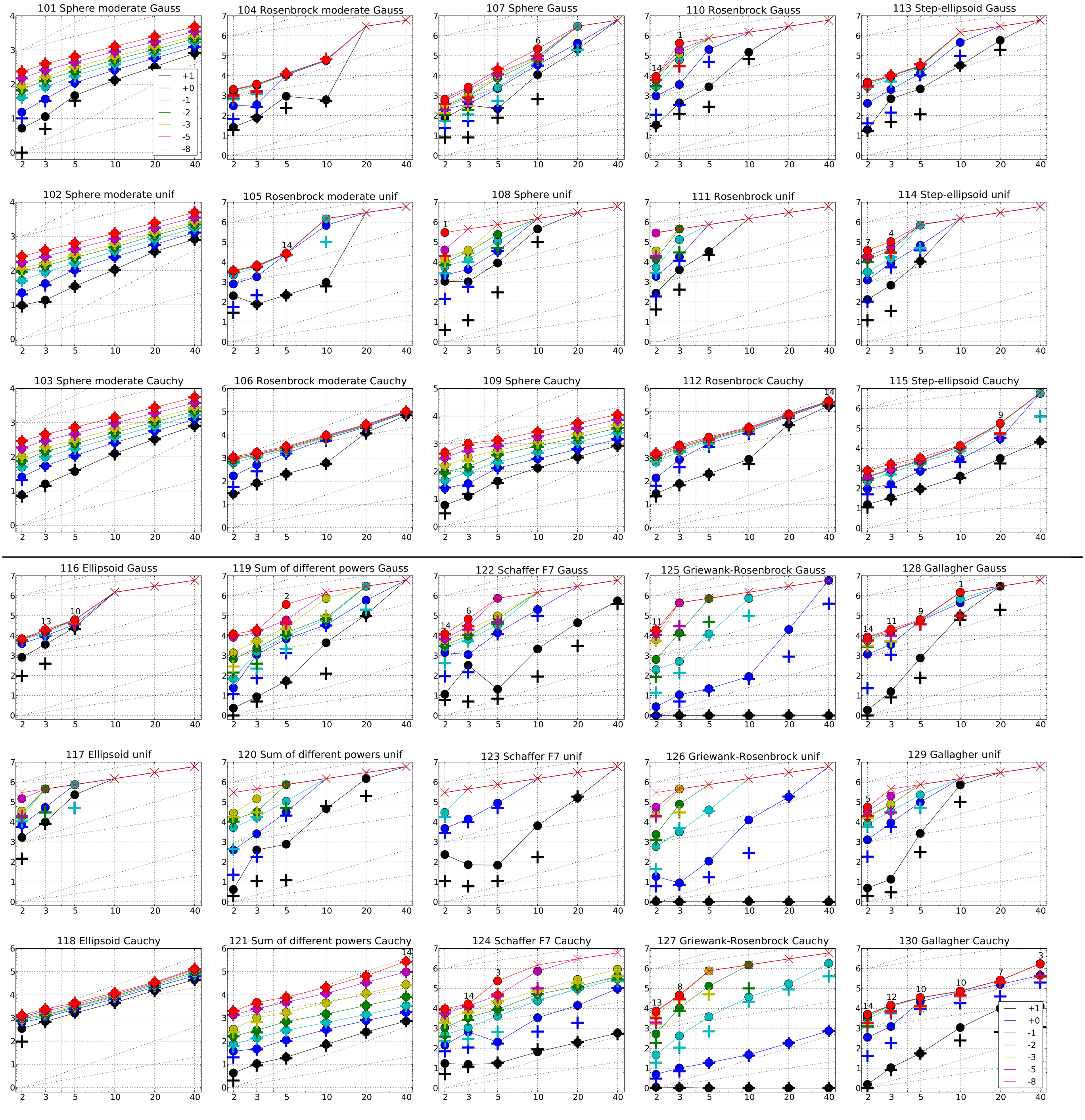

Figure 1: Expected Running Time (ERT, ●) to reach $f_{\mathrm{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_{101} and f_{130}) versus dimension in log-log presentation. The $\mathrm{ERT}(\Delta f)$ equals to $\#\mathrm{FEs}(\Delta f)$ divided by the number of successful trials, where a trial is successful if $f_{\mathrm{opt}} + \Delta f$ was surpassed during the trial. The $\#\mathrm{FEs}(\Delta f)$ are the total number of function evaluations while $f_{\mathrm{opt}} + \Delta f$ was not surpassed during the trial from all respective trials (successful and unsuccessful), and f_{opt} denotes the optimal function value. Crosses ($\times$) indicate the total number of function evaluations $\#\mathrm{FEs}(-\infty)$. Numbers above ERT-symbols indicate the number of successful trials. Annotated numbers on the ordinate are decimal logarithms. Additional grid lines show linear and quadratic scaling.

	f101 in 5-D, N=15, mFE=738					**f101 in 20-D**, N=15, mFE=2654				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	4.6e1	3.7e1	5.6e1	4.6e1	15	3.2e2	3.1e2	3.4e2	3.2e2
1	15	1.1e2	1.0e2	1.2e2	1.1e2	15	5.8e2	5.6e2	6.0e2	5.8e2
1e−1	15	1.8e2	1.7e2	1.9e2	1.8e2	15	8.0e2	7.8e2	8.2e2	8.0e2
1e−3	15	3.2e2	3.1e2	3.3e2	3.2e2	15	1.3e3	1.2e3	1.3e3	1.3e3
1e−5	15	4.4e2	4.3e2	4.5e2	4.4e2	15	1.7e3	1.7e3	1.8e3	1.7e3
1e−8	15	6.4e2	6.2e2	6.6e2	6.4e2	15	2.5e3	2.4e3	2.5e3	2.5e3

	f103 in 5-D, N=15, mFE=1162					**f103 in 20-D**, N=15, mFE=3014				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	3.8e1	3.2e1	4.4e1	3.8e1	15	3.3e2	3.1e2	3.4e2	3.3e2
1	15	1.1e2	9.9e1	1.2e2	1.1e2	15	5.7e2	5.5e2	5.9e2	5.7e2
1e−1	15	1.8e2	1.7e2	1.9e2	1.8e2	15	7.9e2	7.7e2	8.1e2	7.9e2
1e−3	15	3.1e2	3.0e2	3.2e2	3.1e2	15	1.3e3	1.3e3	1.3e3	1.3e3
1e−5	15	4.8e2	4.6e2	5.1e2	4.8e2	15	1.9e3	1.9e3	1.9e3	1.9e3
1e−8	15	7.3e2	6.9e2	7.8e2	7.3e2	15	2.8e3	2.8e3	2.8e3	2.8e3

	f105 in 5-D, N=15, mFE=50008					**f105 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	2.2e2	2.1e2	2.3e2	2.2e2	0	*18e+0*	*14e+0*	*19e+0*	1.3e5
1	14	2.5e4	2.2e4	3.0e4	2.4e4	.	.	.	.	.
1e−1	14	2.6e4	2.2e4	3.0e4	2.4e4	.	.	.	.	.
1e−3	14	2.7e4	2.3e4	3.1e4	2.5e4	.	.	.	.	.
1e−5	14	2.7e4	2.3e4	3.1e4	2.5e4	.	.	.	.	.
1e−8	14	2.7e4	2.3e4	3.2e4	2.5e4	.	.	.	.	.

	f107 in 5-D, N=15, mFE=23686					**f107 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	2.2e2	9.1e1	3.6e2	2.2e2	13	2.1e5	1.8e5	2.5e5	1.8e5
1	15	2.3e3	9.2e2	3.7e3	2.3e3	7	4.2e5	3.3e5	7.4e5	2.0e5
1e−1	15	2.5e3	1.2e3	3.9e3	2.5e3	1	3.0e6	1.5e6	>3e6	2.0e5
1e−3	15	1.0e4	7.8e3	1.3e4	1.0e4	0	*11e−1*	*13e−2*	*14e+0*	2.0e5
1e−5	15	1.3e4	1.0e4	1.6e4	1.3e4	.	.	.	.	.
1e−8	15	2.0e4	1.8e4	2.1e4	2.0e4	.	.	.	.	.

	f109 in 5-D, N=15, mFE=1730					**f109 in 20-D**, N=15, mFE=6338				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	4.5e1	3.6e1	5.5e1	4.5e1	15	3.3e2	3.2e2	3.5e2	3.3e2
1	15	1.3e2	1.2e2	1.5e2	1.3e2	15	6.3e2	6.1e2	6.5e2	6.3e2
1e−1	15	2.2e2	2.0e2	2.4e2	2.2e2	15	1.1e3	1.1e3	1.2e3	1.1e3
1e−3	15	5.7e2	5.2e2	6.2e2	5.7e2	15	2.3e3	2.2e3	2.4e3	2.3e3
1e−5	15	8.7e2	8.1e2	9.4e2	8.7e2	15	3.6e3	3.5e3	3.7e3	3.6e3
1e−8	15	1.4e3	1.3e3	1.4e3	1.4e3	15	5.6e3	5.5e3	5.7e3	5.6e3

	f111 in 5-D, N=15, mFE=50008					**f111 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	12	3.4e4	2.6e4	4.2e4	2.9e4	0	*15e+2*	*52e+1*	*93e+2*	1.8e5
1	0	*46e−1*	*19e−1*	*43e+0*	2.8e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f113 in 5-D, N=15, mFE=37944					**f113 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	2.2e3	8.0e2	3.5e3	2.2e3	5	5.9e5	4.2e5	9.9e5	2.0e5
1	15	1.3e4	8.6e3	1.7e4	1.3e4	0	*21e+0*	*47e−1*	*87e+0*	1.8e5
1e−1	15	2.6e4	2.3e4	3.0e4	2.6e4	.	.	.	.	.
1e−3	15	3.1e4	2.8e4	3.4e4	3.1e4	.	.	.	.	.
1e−5	15	3.1e4	2.8e4	3.4e4	3.1e4	.	.	.	.	.
1e−8	15	3.2e4	2.8e4	3.4e4	3.2e4	.	.	.	.	.

	f115 in 5-D, N=15, mFE=7658					**f115 in 20-D**, N=15, mFE=200074				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	9.1e1	7.7e1	1.1e2	9.1e1	15	3.1e3	2.3e3	3.9e3	3.1e3
1	15	7.1e2	5.9e2	8.3e2	7.1e2	15	3.0e4	2.6e4	3.4e4	3.0e4
1e−1	15	1.8e3	1.4e3	2.3e3	1.8e3	9	1.7e5	1.2e5	2.6e5	9.2e4
1e−3	15	2.6e3	2.1e3	3.1e3	2.6e3	9	1.8e5	1.2e5	2.7e5	9.7e4
1e−5	15	2.6e3	2.1e3	3.0e3	2.6e3	9	1.8e5	1.2e5	2.8e5	9.7e4
1e−8	15	3.4e3	2.8e3	4.0e3	3.4e3	9	1.8e5	1.2e5	2.8e5	9.8e4

	f117 in 5-D, N=15, mFE=50008					**f117 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	3	2.3e5	1.4e5	7.0e5	5.0e4	0	*99e+2*	*53e+2*	*25e+3*	1.8e5
1	1	7.4e5	3.6e5	>7e5	5.0e4	.	.	.	.	.
1e−1	1	7.4e5	3.6e5	>7e5	5.0e4	.	.	.	.	.
1e−3	0	*26e+0*	*39e−1*	*23e+1*	3.5e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f119 in 5-D, N=15, mFE=50008					**f119 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	5.3e1	3.7e1	6.9e1	5.3e1	13	1.1e5	8.3e4	1.5e5	9.8e4
1	15	6.9e3	3.9e3	1.0e4	6.9e3	5	5.9e5	4.2e5	1.0e6	2.0e5
1e−1	15	9.6e3	5.8e3	1.4e4	9.6e3	1	3.0e6	1.5e6	>3e6	2.0e5
1e−3	14	2.6e4	2.1e4	3.0e4	2.6e4	0	*16e−1*	*51e−2*	*10e+0*	1.8e5
1e−5	10	6.1e4	4.9e4	8.0e4	4.1e4	.	.	.	.	.
1e−8	2	3.7e5	1.9e5	>7e5	5.0e4	.	.	.	.	.

	f102 in 5-D, N=15, mFE=754					**f102 in 20-D**, N=15, mFE=2798				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	3.5e1	2.9e1	4.1e1	3.5e1	15	3.6e2	3.4e2	3.8e2	3.6e2
1	15	1.0e2	9.3e1	1.1e2	1.0e2	15	5.8e2	5.6e2	6.0e2	5.8e2
1e−1	15	1.6e2	1.5e2	1.7e2	1.6e2	15	8.2e2	7.9e2	8.5e2	8.2e2
1e−3	15	3.0e2	2.9e2	3.1e2	3.0e2	15	1.3e3	1.3e3	1.3e3	1.3e3
1e−5	15	4.3e2	4.2e2	4.4e2	4.3e2	15	1.8e3	1.7e3	1.8e3	1.8e3
1e−8	15	6.3e2	6.1e2	6.5e2	6.3e2	15	2.5e3	2.5e3	2.5e3	2.5e3

	f104 in 5-D, N=15, mFE=23542					**f104 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	9.3e2	2.3e2	1.6e3	9.3e2	0	*17e+0*	*15e+0*	*18e+0*	1.3e5
1	15	1.1e4	8.4e3	1.3e4	1.1e4	.	.	.	.	.
1e−1	15	1.1e4	9.0e3	1.3e4	1.1e4	.	.	.	.	.
1e−3	15	1.2e4	1.0e4	1.4e4	1.2e4	.	.	.	.	.
1e−5	15	1.2e4	1.0e4	1.5e4	1.2e4	.	.	.	.	.
1e−8	15	1.3e4	1.1e4	1.5e4	1.3e4	.	.	.	.	.

	f106 in 5-D, N=15, mFE=5292					**f106 in 20-D**, N=15, mFE=34994				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	2.1e2	1.9e2	2.2e2	2.1e2	15	1.2e4	1.2e4	1.3e4	1.2e4
1	15	1.6e3	1.3e3	1.9e3	1.6e3	15	2.2e4	2.1e4	2.3e4	2.2e4
1e−1	15	2.2e3	1.9e3	2.5e3	2.2e3	15	2.4e4	2.3e4	2.5e4	2.4e4
1e−3	15	2.7e3	2.4e3	2.9e3	2.7e3	15	2.5e4	2.5e4	2.6e4	2.5e4
1e−5	15	2.9e3	2.6e3	3.2e3	2.9e3	15	2.6e4	2.6e4	2.7e4	2.6e4
1e−8	15	3.2e3	2.9e3	3.4e3	3.2e3	15	2.8e4	2.7e4	2.9e4	2.8e4

	f108 in 5-D, N=15, mFE=50008					**f108 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	9.0e3	4.9e3	1.4e4	9.0e3	0	*72e+0*	*28e+0*	*11e+1*	1.8e5
1	13	3.5e4	2.8e4	4.1e4	3.0e4	.	.	.	.	.
1e−1	6	1.1e5	7.5e4	1.8e5	4.2e4	.	.	.	.	.
1e−3	0	*20e−2*	*58e−4*	*23e−1*	2.2e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f110 in 5-D, N=15, mFE=50008					**f110 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	2.8e3	3.3e2	5.2e3	2.8e3	0	*60e+0*	*39e+0*	*40e+1*	1.8e5
1	3	2.1e5	1.2e5	6.2e5	5.0e4	.	.	.	.	.
1e−1	0	*21e−1*	*70e−2*	*33e−1*	1.1e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f112 in 5-D, N=15, mFE=14628					**f112 in 20-D**, N=15, mFE=137776				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	2.1e2	1.9e2	2.3e2	2.1e2	15	2.9e4	2.6e4	3.1e4	2.9e4
1	15	4.3e3	3.1e3	5.4e3	4.3e3	15	6.4e4	5.5e4	7.3e4	6.4e4
1e−1	15	5.7e3	4.6e3	6.8e3	5.7e3	15	7.0e4	6.1e4	7.9e4	7.0e4
1e−3	15	6.7e3	5.6e3	7.8e3	6.7e3	15	7.4e4	6.5e4	8.3e4	7.4e4
1e−5	15	7.3e3	6.2e3	8.5e3	7.3e3	15	7.6e4	6.7e4	8.5e4	7.6e4
1e−8	15	8.0e3	6.9e3	9.2e3	8.0e3	15	7.9e4	7.1e4	8.8e4	7.9e4

	f114 in 5-D, N=15, mFE=50008					**f114 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.1e4	7.8e3	1.5e4	1.1e4	0	*17e+1*	*92e+0*	*39e+1*	1.8e5
1	8	6.8e4	5.0e4	9.9e4	3.7e4	.	.	.	.	.
1e−1	1	7.0e5	3.3e5	>7e5	5.0e4	.	.	.	.	.
1e−3	0	*99e−2*	*10e−2*	*38e−1*	2.5e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f116 in 5-D, N=15, mFE=50008					**f116 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	2.6e4	2.2e4	3.0e4	2.6e4	0	*13e+2*	*45e+1*	*36e+2*	1.8e5
1	14	3.6e4	3.2e4	4.1e4	3.3e4	.	.	.	.	.
1e−1	10	6.1e4	4.9e4	7.9e4	4.1e4	.	.	.	.	.
1e−3	10	6.2e4	5.0e4	8.0e4	4.2e4	.	.	.	.	.
1e−5	10	6.2e4	5.1e4	8.1e4	4.2e4	.	.	.	.	.
1e−8	10	6.3e4	5.1e4	8.2e4	4.3e4	.	.	.	.	.

	f118 in 5-D, N=15, mFE=5098					**f118 in 20-D**, N=15, mFE=46526				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.6e3	1.4e3	1.8e3	1.6e3	15	1.5e4	1.4e4	1.6e4	1.5e4
1	15	2.4e3	2.2e3	2.6e3	2.4e3	15	2.2e4	2.1e4	2.3e4	2.2e4
1e−1	15	2.8e3	2.7e3	3.0e3	2.8e3	15	2.5e4	2.4e4	2.6e4	2.5e4
1e−3	15	3.3e3	3.2e3	3.5e3	3.3e3	15	3.0e4	2.9e4	3.2e4	3.0e4
1e−5	15	3.8e3	3.6e3	3.9e3	3.8e3	15	3.2e4	3.1e4	3.4e4	3.2e4
1e−8	15	4.5e3	4.3e3	4.6e3	4.5e3	15	3.5e4	3.3e4	3.6e4	3.5e4

	f120 in 5-D, N=15, mFE=50008					**f120 in 20-D**, N=15, mFE=200022				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	7.7e2	3.6e1	1.5e3	7.7e2	2	1.5e6	7.5e5	>3e6	2.0e5
1	12	3.1e4	2.5e4	3.9e4	2.8e4	0	*17e+0*	*66e−1*	*25e+0*	1.8e5
1e−1	6	1.1e5	7.8e4	1.7e5	4.6e4	.	.	.	.	.
1e−3	0	*30e−2*	*23e−3*	*31e−1*	2.0e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 1: Shown are, for functions f_{101}-f_{120} and for a given target difference to the optimal function value Δf: the number of successful trials (#); the expected running time to surpass $f_{\text{opt}} + \Delta f$ (ERT, see Figure 1); the 10%-tile and 90%-tile of the bootstrap distribution of ERT; the average number of function evaluations in successful trials or, if none was successful, as last entry the median number of function evaluations to reach the best function value (RT$_{\text{succ}}$). If $f_{\text{opt}} + \Delta f$ was never reached, figures in *italics* denote the best achieved Δf-value of the median trial and the 10% and 90%-tile trial. Furthermore, N denotes the number of trials, and mFE denotes the maximum of number of function evaluations executed in one trial. See Figure 1 for the names of functions.

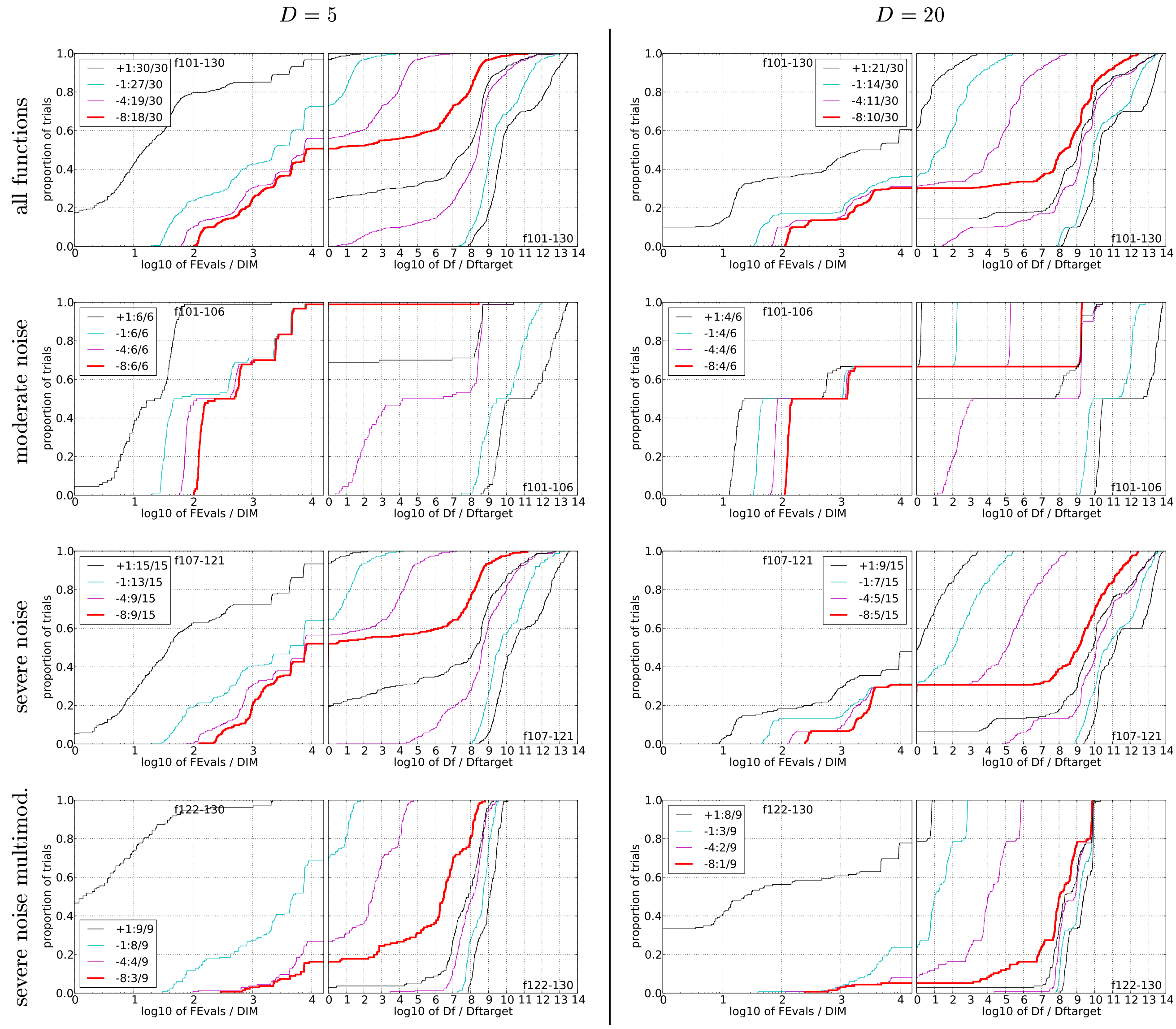

Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left subplots) or versus Δf (right subplots). The thick red line represents the best achieved results. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\rm opt} + \Delta f$ with $\Delta f = 10^k$, where k is the first value in the legend. Right subplots: ECDF of the best achieved Δf divided by 10^k (upper left lines in continuation of the left subplot), and best achieved Δf divided by 10^{-8} for running times of $D, 10\,D, 100\,D \dots$ function evaluations (from right to left cycling black-cyan-magenta). Top row: all results from all functions; second row: moderate noise functions; third row: severe noise functions; fourth row: severe noise and highly-multimodal functions. The legends indicate the number of functions that were solved in at least one trial. FEvals denotes number of function evaluations, D and DIM denote search space dimension, and Δf and Df denote the difference to the optimal function value.

	f121 in 5-D, N=15, mFE=16756					f121 in 20-D, N=15, mFE=74630				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.0e1	1.5e1	2.5e1	2.0e1	15	2.5e2	2.3e2	2.7e2	2.5e2
1	15	1.1e2	9.7e1	1.3e2	1.1e2	15	7.7e2	7.2e2	8.2e2	7.7e2
1e−1	15	2.9e2	2.7e2	3.2e2	2.9e2	15	1.4e3	1.4e3	1.5e3	1.4e3
1e−3	15	1.7e3	1.6e3	1.9e3	1.7e3	15	1.2e4	1.1e4	1.2e4	1.2e4
1e−5	15	4.8e3	4.5e3	5.1e3	4.8e3	15	3.4e4	3.3e4	3.6e4	3.4e4
1e−8	15	7.9e3	7.1e3	8.8e3	7.9e3	15	6.6e4	6.4e4	6.8e4	6.6e4

	f122 in 5-D, N=15, mFE=50008					f122 in 20-D, N=15, mFE=200022				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.1e1	1.1e1	3.2e1	2.1e1	15	4.5e4	2.6e4	6.4e4	4.5e4
1	15	1.4e4	1.0e4	1.8e4	1.4e4	0	*44e−1*	*17e−1*	*62e−1*	1.8e5
1e−1	14	3.3e4	2.8e4	4.0e4	3.1e4	.	.	.	.	.
1e−3	7	9.8e4	7.1e4	1.6e5	4.5e4	.	.	.	.	.
1e−5	1	7.4e5	3.7e5	>7e5	5.0e4	.	.	.	.	.
1e−8	0	*16e−4*	*74e−6*	*70e−3*	4.0e4	.	.	.	.	.

	f123 in 5-D, N=15, mFE=50008					f123 in 20-D, N=15, mFE=200022				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	6.8e1	4.0e1	9.7e1	6.8e1	14	1.6e5	1.4e5	1.9e5	1.5e5
1	7	8.8e4	6.3e4	1.4e5	4.0e4	0	*77e−1*	*49e−1*	*99e−1*	1.8e5
1e−1	0	*10e−1*	*37e−2*	*26e−1*	3.5e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f124 in 5-D, N=15, mFE=50058					f124 in 20-D, N=15, mFE=200086				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.7e1	1.2e1	2.2e1	1.7e1	15	1.9e2	1.7e2	2.1e2	1.9e2
1	15	2.0e2	1.8e2	2.3e2	2.0e2	15	1.4e4	5.2e3	2.3e4	1.4e4
1e−1	15	4.1e3	2.3e3	6.0e3	4.1e3	15	9.1e4	8.0e4	1.0e5	9.1e4
1e−3	15	2.0e4	1.7e4	2.4e4	2.0e4	9	2.8e5	2.2e5	3.9e5	1.6e5
1e−5	13	4.5e4	3.9e4	5.3e4	3.9e4	0	*41e−5*	*37e−6*	*44e−4*	1.8e5
1e−8	3	2.4e5	1.4e5	7.2e5	4.4e4	.	.	.	.	.

	f125 in 5-D, N=15, mFE=50008					f125 in 20-D, N=15, mFE=200022				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	15	2.2e1	1.7e1	2.7e1	2.2e1	15	2.0e4	7.8e3	3.3e4	2.0e4
1e−1	15	1.2e4	8.1e3	1.7e4	1.2e4	0	*59e−2*	*47e−2*	*68e−2*	1.8e5
1e−3	0	*28e−3*	*11e−3*	*70e−3*	4.0e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f126 in 5-D, N=15, mFE=50008					f126 in 20-D, N=15, mFE=200022				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	15	1.1e2	5.4e1	1.7e2	1.1e2	13	1.8e5	1.5e5	2.2e5	1.5e5
1e−1	12	4.0e4	3.0e4	5.3e4	2.8e4	0	*89e−2*	*61e−2*	*11e−1*	1.8e5
1e−3	0	*85e−3*	*47e−3*	*11e−2*	3.5e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f127 in 5-D, N=15, mFE=50058					f127 in 20-D, N=15, mFE=200022				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	15	1.9e1	1.6e1	2.2e1	1.9e1	15	1.8e2	1.5e2	2.1e2	1.8e2
1e−1	15	3.7e3	1.8e3	5.9e3	3.7e3	10	1.7e5	1.3e5	2.3e5	1.2e5
1e−3	1	7.4e5	3.7e5	>7e5	5.0e4	0	*83e−3*	*57e−3*	*15e−2*	8.9e4
1e−5	0	*18e−3*	*16e−4*	*32e−3*	3.5e4	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f128 in 5-D, N=15, mFE=50008					f128 in 20-D, N=15, mFE=200022				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	7.6e2	6.4e1	1.5e3	7.6e2	1	3.0e6	1.5e6	>3e6	2.0e5
1	9	5.5e4	4.1e4	7.8e4	3.6e4	0	*65e+0*	*57e+0*	*71e+0*	1.8e5
1e−1	9	5.9e4	4.6e4	8.1e4	3.8e4	.	.	.	.	.
1e−3	9	6.0e4	4.6e4	8.1e4	3.8e4	.	.	.	.	.
1e−5	9	6.0e4	4.6e4	8.3e4	3.8e4	.	.	.	.	.
1e−8	9	6.4e4	5.0e4	8.9e4	3.9e4	.	.	.	.	.

	f129 in 5-D, N=15, mFE=50008					f129 in 20-D, N=15, mFE=200022				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.7e3	1.4e3	4.1e3	2.7e3	0	*69e+0*	*41e+0*	*73e+0*	1.8e5
1	6	9.8e4	6.9e4	1.6e5	4.1e4	.	.	.	.	.
1e−1	3	2.3e5	1.3e5	7.0e5	4.1e4	.	.	.	.	.
1e−3	0	*18e−1*	*42e−3*	*31e−1*	2.5e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f130 in 5-D, N=15, mFE=50124					f130 in 20-D, N=15, mFE=200086				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	5.5e1	4.3e1	6.7e1	5.5e1	15	9.9e3	5.4e3	1.5e4	9.9e3
1	12	2.2e4	1.4e4	3.1e4	1.7e4	9	1.6e5	1.0e5	2.3e5	1.0e5
1e−1	10	3.3e4	2.4e4	4.5e4	2.7e4	7	2.5e5	1.6e5	4.3e5	9.5e4
1e−3	10	3.4e4	2.4e4	4.5e4	2.7e4	7	2.5e5	1.6e5	4.4e5	9.6e4
1e−5	10	3.4e4	2.4e4	4.6e4	2.7e4	7	2.6e5	1.6e5	4.3e5	9.6e4
1e−8	10	3.5e4	2.5e4	4.6e4	2.8e4	7	2.6e5	1.6e5	4.3e5	9.7e4

Table 2: Shown are, for functions f_{121}-f_{130} and for a given target difference to the optimal function value Δf: the number of successful trials (#); the expected running time to surpass $f_{\rm opt} + \Delta f$ (ERT, see Figure 1); the 10%-tile and 90%-tile of the bootstrap distribution of ERT; the average number of function evaluations in successful trials or, if none was successful, as last entry the median number of function evaluations to reach the best function value ($\mathrm{RT_{succ}}$). If $f_{\rm opt} + \Delta f$ was never reached, figures in *italics* denote the best achieved Δf-value of the median trial and the 10% and 90%-tile trial. Furthermore, N denotes the number of trials, and mFE denotes the maximum of number of function evaluations executed in one trial. See Figure 1 for the names of functions.

with since the performances of the mixed strategy seems to decrease compared to the other noise models. Furthermore, noise affects the scaling of this algorithm since it scales worse on functions f_{107}, f_{108} and f_{109} than on the same function with less noise f_{101}, f_{102} and f_{103}. On the Gauss noise model, which is second most severe, the algorithm can still solve f_{101}. Otherwise, it can only solve functions up to 10-D in the best case.

5. CPU TIMING EXPERIMENT

For the timing experiment, the proposed algorithm was run on f_8 and restarted until at least 30 seconds have passed (according to Figure 2 in [3]). The experiments were conducted with an Intel Core 2 6700 processor (2.66GHz) with Matlab R2008a on Linux 2.6.24.7. The results were 15, 13, 11, 9.7, 9.9, and 13 $\times 10^{-5}$ seconds per function evaluations in dimension 2, 3, 5, 10, 20, and 40 respectively.

6. CONCLUSION

The strategy of mixing CMA-ES with its time and space linear variant results in this algorithm. Tested on the BBOB-2009 noisy functions testbed, it could only solve all ten functions using the less severe Cauchy noise model in 20-D.

7. REFERENCES

[1] 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.

[2] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noisy functions. Technical Report 2009/21, Research Center PPE, 2009.

[3] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA, 2009.

[4] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noisy functions definitions. Technical Report RR-6869, INRIA, 2009.

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

[6] M. Lunacek, D. Whitley, and A. Sutton. The impact of global structure on search. In G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, editors, *PPSN*, volume 5199 of *Lecture Notes in Computer Science*, pages 498–507. Springer, 2008.

[7] R. Ros and N. Hansen. A simple modification in CMA-ES achieving linear time and space complexity. In 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*, pages 296–305. Springer, 2008.