Benchmarking the (1,4)-CMA-ES With Mirrored Sampling and Sequential Selection on the Noisy BBOB-2010 Testbed

[Black-Box Optimization Benchmarking Workshop]

Anne Auger, Dimo Brockhoff, and Nikolaus Hansen
Projet TAO, INRIA Saclay—Ile-de-France
LRI, Bât 490, Univ. Paris-Sud
91405 Orsay Cedex, France
firstname.lastname@inria.fr

ABSTRACT

The Covariance-Matrix-Adaptation Evolution-Strategy (CMA-ES) is a robust stochastic search algorithm for optimizing functions defined on a continuous search space $\mathbb{R}^D$. Recently, mirrored samples and sequential selection have been introduced within CMA-ES to improve its local search performances. In this paper, we benchmark the $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES which implements mirrored samples and sequential selection on the BBOB-2010 noisy testbed. Independent restarts are conducted until a maximal number of $10^4 D$ function evaluations is reached.

Although the tested $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES is only a local search strategy, it solves 8 of the noisy BBOB-2010 functions in 20D and 9 of them in 5D for a target of 10^{-8}. There is also one additional function in 20D and 5 additional functions in 5D where a successful run for at least one of the 15 instances can be reported. Moreover, on 7 of the 8 functions that are solved by the $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES in 20D, we see a large improvement over the best algorithm of the BBOB-2009 benchmarking for the corresponding functions—ranging from an 37% improvement on the sphere with moderate Cauchy noise to a speed-up by a factor of about 3 on the Gallagher function with Cauchy noise.

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

1. INTRODUCTION

Evolution Strategies (ESs) are stochastic search algorithms designed to minimize[1] objective functions, f, mapping a continuous search space $\mathbb{R}^D$ into $\mathbb{R}$. Among ESs, the Covariance-Matrix-Adaptation Evolution-Strategy (CMA-ES) is now a well recognized algorithm. In the standard $(\mu/\mu_w, \lambda)$-CMA-ES [18, 25], at each iteration step n, a set of λ candidate solutions is created by sampling random vectors distributed according to a multivariate normal distribution with mean vector zero and covariance matrix $\boldsymbol{C}_n$. Those λ random vectors denoted $(\mathcal{N}_i(\mathbf{0}, \boldsymbol{C}_n))_{1\leq i\leq\lambda}$ are multiplied by a strictly possitive factor, the step-size σ_n, and added to the current solution $\boldsymbol{X}_n$ to constitute the offspring $\boldsymbol{X}_n^i = \boldsymbol{X}_n + \sigma_n \mathcal{N}_i(\mathbf{0}, \boldsymbol{C}_n)$. After evaluation of the λ offspring, the μ best, i.e., the ones having the smallest objective function values, are selected. The current solution is updated to the average value of the μ best solutions: $\boldsymbol{X}_{n+1} = \sum_{i=1}^{\mu} w_i \boldsymbol{X}_n^{i:\lambda}$, where $w_1 \geq \ldots \geq w_\mu$ and $\sum_{i=1}^{\mu} w_i = 1$ and $\boldsymbol{X}_n^{i:\lambda}$ denotes the i-th best offspring. Covariance matrix and step-size are then updated using solely the information given by the ranking of the offspring. Though originally designed to be a robust local search [26], the $(\mu/\mu_w, \lambda)$-CMA-ES turns out to be also effective for multi-modal functions provided a large enough population size $\mu = \lambda/2$ is chosen [25]. An automatic way to increase the probability to converge on multi-modal functions consists in applying restarts with a successively increasing population size. The strategy is then called IPOP-CMA-ES [13]. However, deceptive functions were constructed for the IPOP-CMA-ES [27, 23]. The BBOB function f_{24} presents, in a highly rugged landscape, on the larger scale an attraction region for the global optimum which is smaller than the one for the local optimum. For that reason, the BIPOP-CMA-ES, combining restarts with increasing population size as well as with a fixed small population size, was proposed [19]. For the large budgets that are needed for most multi-modal problems, the BIPOP-CMA-ES performed overall best for the BBOB-2009 workshop [22].

While BIPOP-CMA-ES was benchmarked, the local search (1+1)-CMA-ES was as well tested [14, 15]. Surprisingly, the (1+1)-variant of CMA-ES could outperform the BIPOP-CMA-ES algorithm by a significant factor on the Gallagher functions f_{21} and f_{22} [11]. On f_{21}, the (1+1)-CMA-ES is 8.2 times (resp. 68.7 times) faster than the BIPOP-CMA-

[1]We assume without loss of generality minimization since maximizing f amounts to minimize $-f$.

ES in dimension 20 (resp. 40); for f_{22}, the (1+1)-CMA-ES is 37 times faster than the BIPOP-CMA-ES in 20D and is able to solve the problem in 40D which the BIPOP-CMA-ES does not allow. However, one major drawback of elitist selection, used in the (1+1)-CMA-ES, is the complete lack of robustness in presence of noise [15].

Motivated by the surprisingly large improvement over the BIPOP-CMA, new *non-elitist* local search ESs have been proposed [6]. Those $(1, \lambda)$-ESs combine a derandomization technique by means of *mirrored samples* with a *sequential selection* scheme. Mirrored samples replace the independent random vectors used for the offspring. Instead of the λ independent random vectors, only $\lambda/2$ (assuming λ is even) independent samples are generated $(\mathcal{N}_{2i-1}(\mathbf{0}, \boldsymbol{C}_n))_{1 \leq i \leq \lambda/2}$. The other $\lambda/2$ samples are replaced by the already generated samples multiplied by -1, i.e., $\mathcal{N}_{2i}(\mathbf{0}, \boldsymbol{C}_n) = -\mathcal{N}_{2i-1}(\mathbf{0}, \boldsymbol{C}_n)$ for all $1 \leq i \leq \lambda/2$. The resulting offspring are two by two symmetrical or *mirrored* with respect to $\boldsymbol{X}_n$. Sequential selection consists in performing the evaluations of the λ offspring sequentially and comparing after each evaluation the offspring solution $\boldsymbol{X}_n^i$ with the current solution $\boldsymbol{X}_n$. If $f(\boldsymbol{X}_n^i) \leq f(\boldsymbol{X}_n)$, the sequence of evaluations is stopped and $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n^i$, saving thus the remaining offspring evaluations.

The impact of mirrored samples and sequential selection has been investigated on the BBOB-2010 for the (1,2)-CMA-ES [2, 3, 7, 8] and for the (1,4)-CMA-ES [4, 5, 9, 10]. The purpose of this paper is to present the results of one of those strategies tested, namely the (1,4)-CMA-ES with mirrored samples and sequential selection on the BBOB-2010 noisy testbed. Since the algorithm tested is a local search strategy, we do not expect that it will perform well on the whole testbed but rather want to see whether the strategy can bring some improvements over last year's results on *certain* functions.

The tested algorithm $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES as well as the CPU timing experiments are described in a complementing paper in the same proceedings [1].

2. RESULTS AND DISCUSSION

Results from experiments according to [21] on the benchmark functions given in [16, 24] are presented in Figures 1, 2 and 3 and in Tables 1, 2 and 3.

Although the tested $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES is only a local search strategy, it solves 8 of the noisy BBOB-2010 functions in 20D and 9 of them in 5D for a target of 10^{-8}. In addition, there is one function in 20D and 5 functions in 5D where a successful run for at least one of the 15 instances can be reported. In the light of this result, it is worth to mention that the noisy test functions in the BBOB-2009 testbed[2] have not been solved as successfully as the noiseless ones: 9 out of the 30 functions could not been solved by any algorithm or solely by the BIPOP-CMA-ES of [20], see [12] for details about the BBOB-2009 results. Moreover, on 7 of the 8 functions that are solved by $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES in 20D, we see even an improvement over the function-wise best algorithm of the BBOB-2009 benchmarking which we detail below.

On the sphere function with moderate noise (f_{103}), the $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES is about 35% better than the best algorithm for this function in 20D and for a target of 10^{-7}. For all other targets, the $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES also outperforms the best algorithm for this function. The best algorithm of the BBOB-2009 benchmarking on this function turns out to be the IPOP-SEP-CMA-ES [28] for small, i.e., difficult targets.

On the Rosenbrock function with moderate Cauchy noise (f_{106}), the $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES outperforms the best BBOB-2009 algorithm for this function in all dimensions (2D, 3D, 5D, 10D, and 20D) and for all small targets which also here is the IPOP-SEP-CMA-ES [28]. The expected running time of the $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES is thereby about 40% smaller than for the IPOP-SEP-CMA-ES for a target value of 10^{-7} in 20D.

Also the sphere function with Cauchy noise (f_{109}) is solved faster by the $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES than the best algorithm of the BBOB-2009 benchmarking on this function in 5D, 10D, and 20D where the improvement is about 50% in 20D and where both the IPOP-SEP-CMA-ES and the BIPOP-CMA-ES [20] are the best algorithms of BBOB-2009 on this function.

A 50% improvement can also be seen on the Rosenbrock function with Cauchy noise (f_{112}) in 20D and for small targets. Better results than the IPOP-SEP-CMA-ES, the best algorithm for this function in BBOB-2009, are also obtained in 2D, 3D, 5D and 10D here.

The improvement over the best algorithm of BBOB-2009 on the ellipsoid function with Cauchy noise (f_{118}) is about 40% in 20D (for all small targets). The best algorithms in BBOB-2009 on this function are the IPOP-SEP-CMA-ES (for a target value of 10^{-7}) as well as the algorithm VNC [17] (for all target values). Better results on f_{118} than the best algorithm of BBOB-2009 are also obtained in 2D, 3D, 5D, and 10D.

On the sum of different powers function comprising Cauchy noise (f_{121}), the $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES shows expected running times that are at least 48% lower than the best algorithm of BBOB-2009 on this function in 20D and for small target values. Better results than the best BBOB-2009 algorithm on this function in 2D, 3D, 5D, and 10D can also be reported.

Last, also on the Gallagher function with Cauchy noise (f_{130}), an improvement over the best algorithm of BBOB-2009 on this function can be seen in the results. Here, we see the largest impact of the $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES, where the expected running times are, in 20D and for several small targets, by a factor of about 3 smaller than the ones of the best algorithm of BBOB-2009 on this function. Also the results in 2D, 3D, 5D, and 10D are better for the $(1,4_{\mathrm{m}}^{\mathrm{s}})$-CMA-ES where the improvement factor only slightly differs in 3D, 5D, and 10D. At least for small targets and dimension ≥ 5, the IPOP-SEP-CMA-ES is here also the best algorithm of BBOB-2009.

Note that all functions, where an improvement over the best algorithm of the BBOB-2009 benchmarking can be reported, comprise a Cauchy noise. Cauchy noise is only sampled 20% of the time, such that it is enough to be robust to postive and negative outliers for solving those functions. For the other noise types, most probably a larger population size or another method to cope with the noise is needed. Furthermore, the maximum number of function evaluations was chosen quite small for solving the more difficult noise types up to the final target value.

[2]These are the same functions than in the BBOB-2010 testbed with the only difference that instead of 15 instances per function, three independent runs were performed on 5 different instances within BBOB-2009.

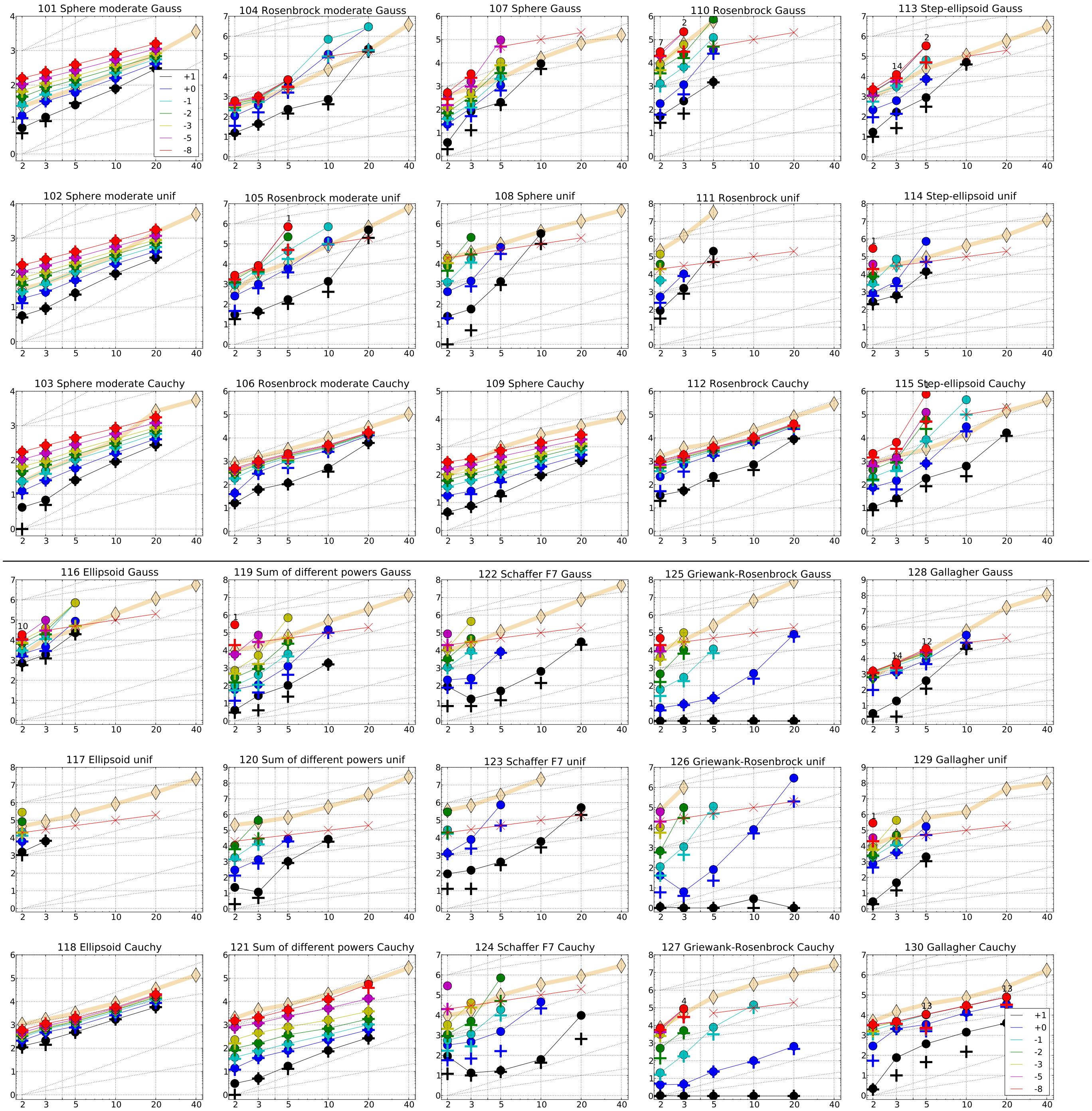

Figure 1: Expected Running Time (ERT, ●) to reach $f_{\mathrm{opt}} + \Delta f$ and median number of f-evaluations from successful trials (+), for $\Delta f = 10^{\{+1,0,-1,-2,-3,-5,-8\}}$ (the exponent is given in the legend of f_{101} and f_{130}) versus dimension in log-log presentation. For each function and dimension, $\mathrm{ERT}(\Delta f)$ 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. The $\#\mathrm{FEs}(\Delta f)$ are the total number (sum) of f-evaluations while $f_{\mathrm{opt}} + \Delta f$ was not surpassed in the trial, from all (successful and unsuccessful) trials, and f_{opt} is the optimal function value. Crosses (×) indicate the total number of f-evaluations, $\#\mathrm{FEs}(-\infty)$, divided by the number of trials. Numbers above ERT-symbols indicate the number of successful trials. Y-axis annotations are decimal logarithms. The thick light line with diamonds shows the single best results from BBOB-2009 for $\Delta f = 10^{-8}$. Additional grid lines show linear and quadratic scaling.

Δf	**f_{101} in 5-D**, N=15, mFE=450					**f_{101} in 20-D**, N=15, mFE=1823				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.7e1	3.0e0	4.6e1	2.7e1	15	3.0e2	2.5e2	3.5e2	3.0e2
1	15	6.1e1	2.7e1	8.2e1	6.1e1	15	4.3e2	3.7e2	5.0e2	4.3e2
1e−1	15	1.0e2	7.2e1	1.2e2	1.0e2	15	5.6e2	4.9e2	6.6e2	5.6e2
1e−3	15	1.8e2	1.5e2	2.0e2	1.8e2	15	8.4e2	7.7e2	9.3e2	8.4e2
1e−5	15	2.7e2	2.1e2	3.1e2	2.7e2	15	1.1e3	1.1e3	1.2e3	1.1e3
1e−8	15	3.9e2	3.7e2	4.5e2	3.9e2	15	1.6e3	1.4e3	1.7e3	1.6e3

Δf	**f_{102} in 5-D**, N=15, mFE=524					**f_{102} in 20-D**, N=15, mFE=2063				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.5e1	7.0e0	4.8e1	2.5e1	15	2.7e2	2.2e2	3.2e2	2.7e2
1	15	6.1e1	3.3e1	8.0e1	6.1e1	15	4.2e2	3.6e2	4.7e2	4.2e2
1e−1	15	1.1e2	8.2e1	1.3e2	1.1e2	15	5.7e2	4.7e2	6.8e2	5.7e2
1e−3	15	1.9e2	1.5e2	2.1e2	1.9e2	15	8.9e2	7.8e2	1.0e3	8.9e2
1e−5	15	2.7e2	2.3e2	3.0e2	2.7e2	15	1.2e3	1.1e3	1.3e3	1.2e3
1e−8	15	4.1e2	3.5e2	5.0e2	4.1e2	15	1.8e3	1.6e3	1.9e3	1.8e3

Δf	**f_{103} in 5-D**, N=15, mFE=521					**f_{103} in 20-D**, N=15, mFE=1917				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.6e1	2.0e0	4.5e1	2.6e1	15	2.8e2	2.0e2	3.8e2	2.8e2
1	15	5.8e1	3.1e1	8.9e1	5.8e1	15	4.1e2	3.5e2	5.0e2	4.1e2
1e−1	15	1.0e2	8.3e1	1.3e2	1.0e2	15	5.8e2	4.8e2	6.8e2	5.8e2
1e−3	15	1.9e2	1.4e2	2.4e2	1.9e2	15	8.8e2	7.9e2	9.8e2	8.8e2
1e−5	15	2.8e2	2.5e2	3.2e2	2.8e2	15	1.2e3	1.0e3	1.4e3	1.2e3
1e−8	15	4.4e2	3.7e2	4.7e2	4.4e2	15	1.7e3	1.6e3	1.9e3	1.7e3

Δf	**f_{104} in 5-D**, N=15, mFE=25439					**f_{104} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.3e2	7.3e1	8.7e2	2.3e2	9	2.3e5	1.5e4	4.4e5	9.8e4
1	15	3.3e3	5.0e2	8.3e3	3.3e3	1	3.0e6	4.7e5	5.2e6	1.7e5
1e−1	15	6.4e3	8.6e2	1.7e4	6.4e3	1	3.0e6	3.7e5	6.0e6	1.7e5
1e−3	15	6.7e3	1.2e3	2.0e4	6.7e3	0	*91e−1*	*19e−1*	*12e+0*	1.1e5
1e−5	15	6.8e3	1.8e3	2.0e4	6.8e3	.	.	.	.	.
1e−8	15	7.0e3	1.5e3	2.0e4	7.0e3	.	.	.	.	.

Δf	**f_{105} in 5-D**, N=15, mFE=50004					**f_{105} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.7e2	8.3e1	3.3e2	1.7e2	5	5.0e5	7.5e4	1.2e6	1.0e5
1	15	6.2e3	2.1e2	1.2e4	6.2e3	0	*13e+0*	*91e−1*	*16e+0*	8.5e4
1e−1	9	4.5e4	3.4e3	1.1e5	1.2e4	.	.	.	.	.
1e−3	1	7.2e5	6.7e4	1.4e6	1.7e4	.	.	.	.	.
1e−5	1	7.2e5	6.7e4	1.4e6	1.7e4	.	.	.	.	.
1e−8	1	7.2e5	6.7e4	1.5e6	1.7e4	.	.	.	.	.

Δf	**f_{106} in 5-D**, N=15, mFE=4228					**f_{106} in 20-D**, N=15, mFE=34780				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.2e2	7.6e1	1.9e2	1.2e2	15	6.3e3	3.6e3	8.5e3	6.3e3
1	15	1.0e3	3.3e2	2.8e3	1.0e3	15	1.3e4	4.2e3	1.7e4	1.3e4
1e−1	15	1.5e3	7.4e2	3.1e3	1.5e3	15	1.4e4	6.6e3	1.9e4	1.4e4
1e−3	15	1.8e3	1.0e3	3.5e3	1.8e3	15	1.6e4	8.0e3	2.0e4	1.6e4
1e−5	15	2.0e3	1.2e3	3.6e3	2.0e3	15	1.6e4	8.5e3	2.1e4	1.6e4
1e−8	15	2.1e3	1.4e3	3.7e3	2.1e3	15	1.7e4	9.1e3	2.1e4	1.7e4

Δf	**f_{107} in 5-D**, N=15, mFE=50004					**f_{107} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.0e2	1.1e1	5.1e2	2.0e2	0	*56e+0*	*40e+0*	*76e+0*	1.5e5
1	15	1.1e3	1.5e2	3.2e3	1.1e3	.	.	.	.	.
1e−1	15	2.3e3	3.1e2	5.5e3	2.3e3	.	.	.	.	.
1e−3	15	1.1e4	6.8e2	2.5e4	1.1e4	.	.	.	.	.
1e−5	6	9.6e4	1.8e4	2.4e5	2.1e4	.	.	.	.	.
1e−8	0	*15e−6*	*59e−8*	*30e−5*	2.0e4	.	.	.	.	.

Δf	**f_{108} in 5-D**, N=15, mFE=50004					**f_{108} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.3e3	1.1e1	3.2e3	1.3e3	0	*95e+0*	*75e+0*	*11e+1*	1.4e5
1	8	6.8e4	2.4e4	1.7e5	2.4e4	.	.	.	.	.
1e−1	0	*94e−2*	*34e−2*	*21e−1*	2.4e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{109} in 5-D**, N=15, mFE=975					**f_{109} in 20-D**, N=15, mFE=3250				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.1e1	6.0e0	3.0e1	2.1e1	15	3.0e2	2.4e2	3.8e2	3.0e2
1	15	6.5e1	3.7e1	9.2e1	6.5e1	15	5.1e2	4.1e2	6.1e2	5.1e2
1e−1	15	1.2e2	7.3e1	1.8e2	1.2e2	15	7.6e2	6.0e2	8.6e2	7.6e2
1e−3	15	2.8e2	2.3e2	3.4e2	2.8e2	15	1.3e3	1.2e3	1.4e3	1.3e3
1e−5	15	4.6e2	3.8e2	5.9e2	4.6e2	15	1.9e3	1.7e3	2.3e3	1.9e3
1e−8	15	7.2e2	5.6e2	8.6e2	7.2e2	15	2.7e3	2.5e3	2.9e3	2.7e3

Δf	**f_{110} in 5-D**, N=15, mFE=50004					**f_{110} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.5e3	9.3e1	3.1e3	1.5e3	0	*55e+3*	*44e+3*	*11e+4*	8.1e4
1	11	3.7e4	3.0e3	8.4e4	1.8e4	.	.	.	.	.
1e−1	5	1.2e5	3.0e3	2.8e5	2.1e4	.	.	.	.	.
1e−3	0	*28e−2*	*30e−3*	*16e−1*	3.0e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{111} in 5-D**, N=15, mFE=50004					**f_{111} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	3	2.0e5	3.9e3	5.0e5	3.3e3	0	*70e+3*	*56e+3*	*11e+4*	8.9e4
1	0	*22e+0*	*80e−1*	*95e+0*	2.3e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{112} in 5-D**, N=15, mFE=6622					**f_{112} in 20-D**, N=15, mFE=64245				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.3e2	9.1e1	3.0e2	2.3e2	15	8.9e3	3.9e3	1.3e4	8.9e3
1	15	1.9e3	5.5e2	4.0e3	1.9e3	15	3.1e4	1.4e4	5.1e4	3.1e4
1e−1	15	2.8e3	1.1e3	4.6e3	2.8e3	15	3.5e4	2.1e4	5.3e4	3.5e4
1e−3	15	3.4e3	1.9e3	5.1e3	3.4e3	15	3.7e4	2.3e4	5.6e4	3.7e4
1e−5	15	3.7e3	2.2e3	5.4e3	3.7e3	15	3.8e4	2.5e4	5.7e4	3.8e4
1e−8	15	4.0e3	2.4e3	5.8e3	4.0e3	15	4.0e4	2.2e4	5.8e4	4.0e4

Δf	**f_{113} in 5-D**, N=15, mFE=50004					**f_{113} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	8.9e2	7.8e1	1.8e3	8.9e2	0	*18e+1*	*12e+1*	*23e+1*	1.4e5
1	15	7.5e3	5.4e2	1.4e4	7.5e3	.	.	.	.	.
1e−1	8	6.6e4	7.7e3	2.0e5	2.2e4	.	.	.	.	.
1e−3	2	3.4e5	5.8e4	8.6e5	1.1e4	.	.	.	.	.
1e−5	2	3.4e5	6.4e4	8.4e5	1.1e4	.	.	.	.	.
1e−8	2	3.4e5	1.5e4	8.4e5	1.4e4	.	.	.	.	.

Δf	**f_{114} in 5-D**, N=15, mFE=50003					**f_{114} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	14	1.4e4	3.4e3	2.2e4	1.1e4	0	*53e+1*	*34e+1*	*84e+1*	1.4e5
1	1	7.3e5	8.4e4	1.7e6	3.4e4	.	.	.	.	.
1e−1	0	*21e−1*	*11e−1*	*93e−1*	2.6e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{115} in 5-D**, N=15, mFE=50004					**f_{115} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.8e2	4.7e1	4.1e2	1.8e2	15	1.7e4	4.6e3	4.9e4	1.7e4
1	15	8.0e2	9.5e1	1.4e3	8.0e2	0	*45e−1*	*30e−1*	*59e−1*	8.3e4
1e−1	15	8.4e3	2.9e3	1.9e4	8.4e3	.	.	.	.	.
1e−3	5	1.2e5	1.5e4	2.4e5	2.4e4	.	.	.	.	.
1e−5	5	1.2e5	1.4e4	2.6e5	2.4e4	.	.	.	.	.
1e−8	1	7.5e5	9.9e4	1.8e6	4.9e4	.	.	.	.	.

Δf	**f_{116} in 5-D**, N=15, mFE=50004					**f_{116} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	13	2.3e4	8.5e2	5.3e4	1.5e4	0	*23e+3*	*19e+3*	*33e+3*	5.0e4
1	6	8.7e4	4.3e3	1.6e5	1.2e4	.	.	.	.	.
1e−1	1	7.2e5	1.2e5	1.9e6	1.5e4	.	.	.	.	.
1e−3	1	7.2e5	1.6e4	1.9e6	1.6e4	.	.	.	.	.
1e−5	0	*28e−1*	*28e−2*	*17e+0*	2.0e4	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{117} in 5-D**, N=15, mFE=50004					**f_{117} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	0	*13e+1*	*51e+0*	*20e+1*	2.6e4	0	*32e+3*	*18e+3*	*40e+3*	8.7e4
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{118} in 5-D**, N=15, mFE=3000					**f_{118} in 20-D**, N=15, mFE=22481				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	4.9e2	3.4e2	9.6e2	4.9e2	15	5.7e3	4.4e3	6.8e3	5.7e3
1	15	8.7e2	5.7e2	1.3e3	8.7e2	15	9.5e3	6.9e3	1.3e4	9.5e3
1e−1	15	1.2e3	7.3e2	1.6e3	1.2e3	15	1.3e4	9.7e3	1.5e4	1.3e4
1e−3	15	1.5e3	1.0e3	1.8e3	1.5e3	15	1.7e4	1.5e4	1.8e4	1.7e4
1e−5	15	1.7e3	1.2e3	2.0e3	1.7e3	15	1.8e4	1.7e4	2.0e4	1.8e4
1e−8	15	2.0e3	1.6e3	3.0e3	2.0e3	15	2.0e4	1.8e4	2.2e4	2.0e4

Δf	**f_{119} in 5-D**, N=15, mFE=50004					**f_{119} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.0e2	3.0e0	5.0e2	1.0e2	0	*13e+0*	*11e+0*	*18e+0*	1.4e5
1	15	1.3e3	1.1e2	3.4e3	1.3e3	.	.	.	.	.
1e−1	15	6.4e3	7.0e2	1.6e4	6.4e3	.	.	.	.	.
1e−3	1	7.1e5	8.3e4	1.8e6	7.8e3	.	.	.	.	.
1e−5	0	*55e−4*	*12e−4*	*15e−3*	2.1e4	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{120} in 5-D**, N=15, mFE=50004					**f_{120} in 20-D**, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.0e3	2.0e0	2.7e3	1.0e3	0	*35e+0*	*22e+0*	*40e+0*	9.8e4
1	13	2.8e4	5.4e3	6.6e4	2.0e4	.	.	.	.	.
1e−1	0	*69e−2*	*46e−2*	*13e−1*	3.5e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
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 $f_{\mathrm{opt}} + \Delta f$ (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_{succ}). If $f_{\mathrm{opt}} + \Delta f$ 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.

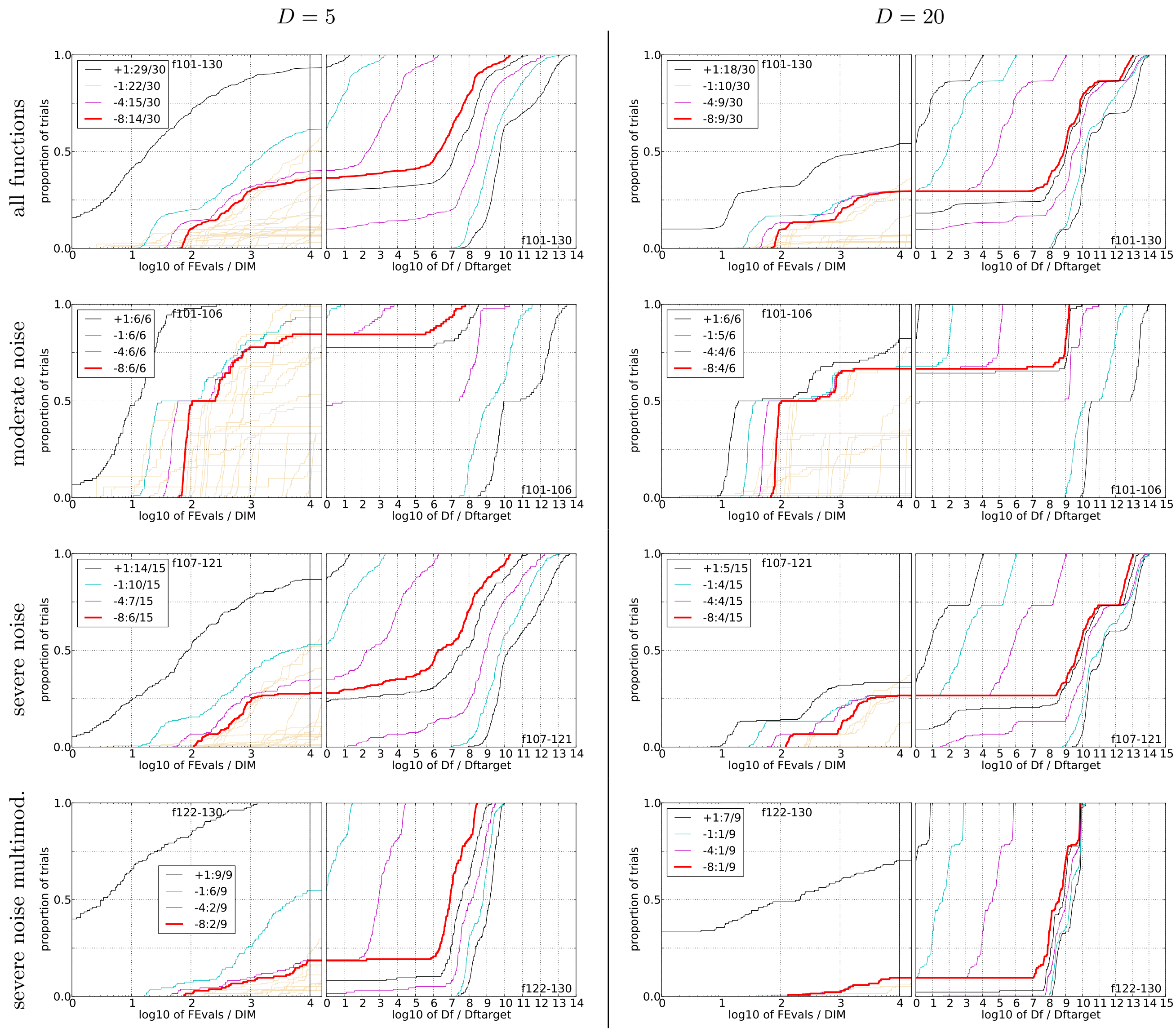

Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left subplots) or versus Δf (right subplots). The thick red line represents the best achieved results. Left subplots: ECDF of the running time (number of function evaluations), divided 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 $D, 10\,D, 100\,D\ldots$ function evaluations (from right to left cycling black-cyan-magenta). 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. Light brown lines in the background show ECDFs for target value 10^{-8} of all algorithms benchmarked during BBOB-2009.

Δf	f_{121} in 5-D, N=15, mFE=7754					f_{121} in 20-D, N=15, mFE=185451				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.7e1	3.0e0	3.3e1	1.7e1	15	2.8e2	2.0e2	3.8e2	2.8e2
1	15	8.0e1	2.6e1	1.0e2	8.0e1	15	6.1e2	4.9e2	7.3e2	6.1e2
1e−1	15	1.6e2	1.0e2	2.1e2	1.6e2	15	1.0e3	8.1e2	1.1e3	1.0e3
1e−3	15	8.1e2	5.6e2	1.2e3	8.1e2	15	3.8e3	3.2e3	4.5e3	3.8e3
1e−5	15	2.3e3	1.7e3	3.0e3	2.3e3	15	1.3e4	1.2e4	1.6e4	1.3e4
1e−8	15	4.4e3	3.2e3	5.2e3	4.4e3	15	5.5e4	3.4e4	1.2e5	5.5e4

Δf	f_{123} in 5-D, N=15, mFE=50004					f_{123} in 20-D, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	4.3e2	2.0e0	7.4e2	4.3e2	5	5.2e5	9.6e4	1.3e6	1.2e5
1	1	7.4e5	9.4e4	1.8e6	4.4e4	0	*11e+0*	*92e−1*	*14e+0*	9.7e4
1e−1	0	*18e−1*	*16e−1*	*23e−1*	2.6e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{125} in 5-D, N=15, mFE=50004					f_{125} in 20-D, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	15	2.0e1	4.5e0	3.9e1	2.0e1	14	8.2e4	2.8e3	1.4e5	6.8e4
1e−1	14	1.2e4	5.5e2	2.8e4	8.5e3	0	*95e−2*	*82e−2*	*98e−2*	9.8e4
1e−3	0	*54e−3*	*29e−3*	*94e−3*	1.4e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{127} in 5-D, N=15, mFE=50004					f_{127} in 20-D, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	15	2.4e1	3.0e0	4.7e1	2.4e1	15	6.5e2	3.2e2	1.3e3	6.5e2
1e−1	14	7.8e3	1.6e3	1.1e4	4.2e3	0	*23e−2*	*13e−2*	*53e−2*	1.4e5
1e−3	0	*42e−3*	*24e−3*	*90e−3*	1.6e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{129} in 5-D, N=15, mFE=50004					f_{129} in 20-D, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.1e3	3.8e1	5.7e3	2.1e3	0	*72e+0*	*68e+0*	*74e+0*	8.9e4
1	4	1.7e5	4.1e4	3.9e5	3.4e4	.	.	.	.	.
1e−1	0	*19e−1*	*39e−2*	*25e−1*	2.3e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{122} in 5-D, N=15, mFE=50004					f_{122} in 20-D, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	5.0e1	7.0e0	4.8e1	5.0e1	15	3.0e4	2.2e3	7.7e4	3.0e4
1	15	8.3e3	1.7e3	1.9e4	8.3e3	0	*75e−1*	*63e−1*	*85e−1*	8.2e4
1e−1	0	*35e−2*	*18e−2*	*44e−2*	3.8e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{124} in 5-D, N=15, mFE=50004					f_{124} in 20-D, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.7e1	1.0e0	3.3e1	1.7e1	15	9.7e3	1.7e2	5.1e4	9.7e3
1	15	1.5e3	8.0e1	6.7e3	1.5e3	0	*40e−1*	*24e−1*	*49e−1*	1.3e5
1e−1	13	1.9e4	3.9e2	5.4e4	1.1e4	.	.	.	.	.
1e−3	0	*46e−3*	*14e−3*	*11e−2*	1.6e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{126} in 5-D, N=15, mFE=50004					f_{126} in 20-D, N=15, mFE=200003				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	15	8.2e1	1.0e0	4.4e2	8.2e1	1	2.9e6	3.2e5	7.2e6	1.2e5
1e−1	5	1.1e5	6.8e3	2.4e5	1.4e4	0	*13e−1*	*11e−1*	*15e−1*	9.5e4
1e−3	0	*12e−2*	*84e−3*	*16e−2*	1.8e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{128} in 5-D, N=15, mFE=50001					f_{128} in 20-D, N=15, mFE=200004				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	3.8e2	2.8e1	8.6e2	3.8e2	0	*65e+0*	*43e+0*	*69e+0*	9.2e4
1	15	7.9e3	2.0e2	2.0e4	7.9e3	.	.	.	.	.
1e−1	15	1.7e4	8.3e1	3.8e4	1.7e4	.	.	.	.	.
1e−3	13	2.7e4	7.6e3	8.3e4	1.9e4	.	.	.	.	.
1e−5	13	3.2e4	1.4e4	6.2e4	2.4e4	.	.	.	.	.
1e−8	12	4.2e4	2.0e4	8.7e4	3.0e4	.	.	.	.	.

Δf	f_{130} in 5-D, N=15, mFE=50003					f_{130} in 20-D, N=15, mFE=200003				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	3.7e2	1.7e1	1.6e3	3.7e2	15	4.1e3	4.4e2	1.2e4	4.1e3
1	15	3.4e3	8.3e1	1.6e4	3.4e3	15	3.5e4	5.1e2	8.9e4	3.5e4
1e−1	13	1.1e4	1.0e2	5.1e4	2.9e3	13	7.8e4	4.6e3	1.3e5	4.7e4
1e−3	13	1.1e4	1.9e2	5.0e4	3.0e3	13	7.8e4	5.1e3	1.4e5	4.7e4
1e−5	13	1.1e4	2.9e2	5.0e4	3.1e3	13	7.9e4	5.6e3	1.7e5	4.8e4
1e−8	13	1.1e4	4.0e2	5.1e4	3.3e3	13	7.9e4	6.3e3	2.1e5	4.9e4

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 (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.

Acknowledgments

This work receives support by the French national research agency (ANR) within the SYSCOMM project ANR-08-SYSC-017 and within the COSINUS project ANR-08-COSI-007-12.

3. REFERENCES

[1] A. Auger, D. Brockhoff, and N. Hansen. Benchmarking the (1,4)-CMA-ES with mirrored sampling and sequential selection on the noiseless BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[2] A. Auger, D. Brockhoff, and N. Hansen. Investigating the impact of sequential selection in the (1,2)-CMA-ES on the noiseless BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[3] A. Auger, D. Brockhoff, and N. Hansen. Investigating the impact of sequential selection in the (1,2)-CMA-ES on the noisy BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[4] A. Auger, D. Brockhoff, and N. Hansen. Investigating the impact of sequential selection in the (1,4)-CMA-ES on the noiseless BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[5] A. Auger, D. Brockhoff, and N. Hansen. Investigating the impact of sequential selection in the (1,4)-CMA-ES on the noisy BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[6] A. Auger, D. Brockhoff, and N. Hansen. Mirrored sampling and sequential selection for evolution

Table 3: ERT **loss ratio (see Figure 3) compared to the respective best result from BBOB-2009 for budgets given in the first column. The last row RL_{US}/D gives the number of function evaluations in unsuccessful runs divided by dimension. Shown are the smallest, 10%-ile, 25%-ile, 50%-ile, 75%-ile and 90%-ile value (smaller values are better).**

#FEs/D	f_{101}–f_{130} in 5-D, maxFE/D=10000					
	best	10%	25%	**med**	75%	90%
2	0.91	1.8	2.7	6.2	10	10
10	0.94	1.4	1.9	5.2	36	50
100	0.52	1.1	2.1	5.4	43	3.0e2
1e3	0.60	0.67	3.8	5.8	26	2.5e3
1e4	0.32	0.66	1.8	12	24	2.5e4
RL_{US}/D	1e4	1e4	1e4	1e4	1e4	1e4

#FEs/D	f_{101}–f_{130} in 20-D, maxFE/D=10000					
	best	10%	25%	**med**	75%	90%
2	1.0	5.0	40	40	40	40
10	1.0	3.0	4.7	1.6e2	2.0e2	2.0e2
100	0.52	0.65	1.4	1.6e2	2.0e3	2.0e3
1e3	0.27	0.46	1.2	3.7e2	2.0e4	2.0e4
1e4	0.31	0.55	1.2	2.8e2	2.4e3	2.0e5
1e5	0.31	0.55	1.2	1.8e3	4.7e3	2.0e6
RL_{US}/D	1e4	1e4	1e4	1e4	1e4	1e4

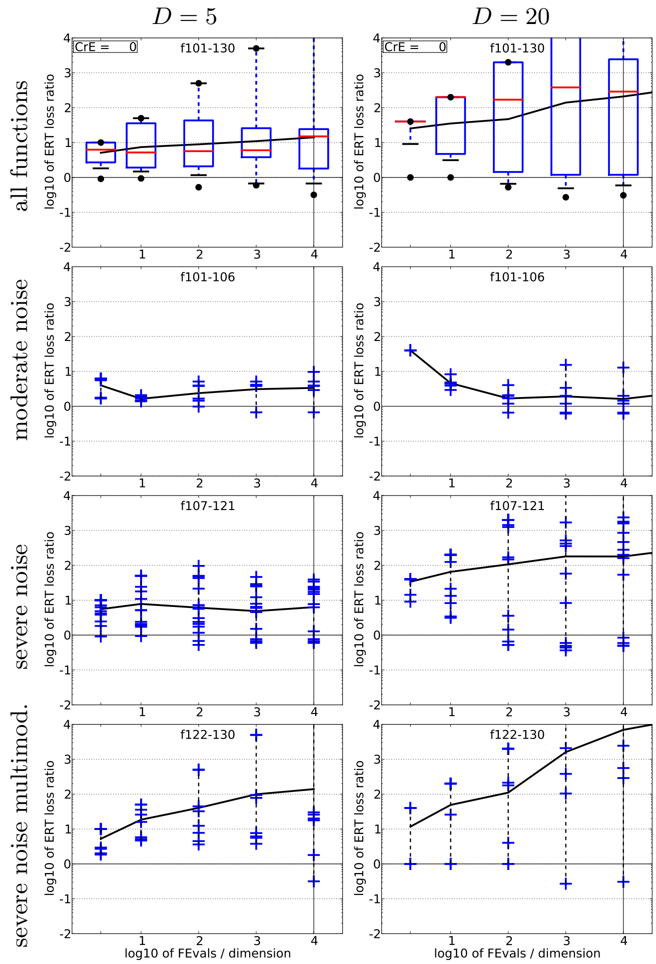

Figure 3: ERT loss ratio versus given budget FEvals. The target value f_t for ERT (see Figure 1) is the smallest (best) recorded function value such that $\mathrm{ERT}(f_t) \leq \mathrm{FEvals}$ for the presented algorithm. Shown is FEvals divided by the respective best $\mathrm{ERT}(f_t)$ from BBOB-2009 for functions f_{101}–f_{130} in 5-D and 20-D. Each ERT is multiplied by $\exp(\mathrm{CrE})$ correcting for the parameter crafting effort. Line: geometric mean. Box-Whisker error bar: 25-75%-ile with median (box), 10-90%-ile (caps), and minimum and maximum ERT loss ratio (points). The vertical line gives the maximal number of function evaluations in this function subset.

strategies. Rapport de Recherche RR-7249, INRIA Saclay—Île-de-France, April 2010.

[7] A. Auger, D. Brockhoff, and N. Hansen. Mirrored variants of the (1,2)-CMA-ES compared on the noiseless BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[8] A. Auger, D. Brockhoff, and N. Hansen. Mirrored variants of the (1,2)-CMA-ES compared on the noisy BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[9] A. Auger, D. Brockhoff, and N. Hansen. Mirrored variants of the (1,4)-CMA-ES compared on the noiseless BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[10] A. Auger, D. Brockhoff, and N. Hansen. Mirrored variants of the (1,4)-CMA-ES compared on the noisy BBOB-2010 testbed. In *GECCO (Companion)*, 2010.

[11] A. Auger, S. Finck, N. Hansen, and R. Ros. BBOB 2009: Comparison tables of all algorithms on all noiseless functions. Technical Report RT-0383, INRIA, April 2010.

[12] A. Auger, S. Finck, N. Hansen, and R. Ros. BBOB 2009: Comparison tables of all algorithms on all noisy functions. Technical Report RT-0384, INRIA, 04 2010.

[13] 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.

[14] A. Auger and N. Hansen. Benchmarking the (1+1)-CMA-ES on the BBOB-2009 function testbed. In Rothlauf [29], pages 2459–2466.

[15] A. Auger and N. Hansen. Benchmarking the (1+1)-CMA-ES on the BBOB-2009 noisy testbed. In Rothlauf [29], pages 2467–2472.

[16] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2010: Presentation of the noisy functions. Technical Report 2009/21, Research Center PPE, 2010.

[17] C. García-Martínez and M. Lozano. A continuous variable neighbourhood search based on specialised EAs: application to the noisy BBO-benchmark 2009 testbed. In Rothlauf [29], pages 2367–2374.

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

[19] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In Rothlauf [29], pages 2389–2396.

[20] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 noisy testbed. In Rothlauf [29], pages 2397–2402.

[21] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Technical Report RR-7215, INRIA, 2010.

[22] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2010)*. ACM Press, 2010. to appear.

[23] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009.

[24] 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. Updated February 2010.

[25] 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.

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

[27] M. Lunacek, D. Whitley, and A. Sutton. The impact of global structure on search. In *Proceedings of the 10th international conference on Parallel Problem Solving from Nature*, pages 498–507, Berlin, Heidelberg, 2008. Springer-Verlag.

[28] R. Ros. Benchmarking sep-CMA-ES on the BBOB-2009 noisy testbed. In Rothlauf [29], pages 2441–2446.

[29] F. Rothlauf, editor. *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009, Companion Material*. ACM, 2009.