Mirrored Variants of the (1,4)-CMA-ES Compared on the Noiseless BBOB-2010 Testbed

Anne Auger, Dimo Brockhoff, Nikolaus Hansen

Mirrored Variants of the (1,4)-CMA-ES Compared on the Noiseless 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

Derandomization by means of mirrored samples has been recently introduced to enhance the performances of $(1, \lambda)$-Evolution-Strategies (ESs) with the aim of designing fast and robust stochastic local search algorithms. This paper compares on the BBOB-2010 noiseless benchmark testbed two variants of the (1,4)-CMA-ES where the mirrored samples are used. Independent restarts are conducted up to a total budget of $10^4 D$ function evaluations, where D is the dimension of the search space.

The results show that the improved variants are significantly faster than the baseline (1,4)-CMA-ES on 4 functions in 20D (respectively 7 when using sequential selection in addition) by a factor of up to 3 (on the attractive sector function). In no case, the (1,4)-CMA-ES is significantly faster on any tested target function value in 5D and 20D. Moreover, the algorithm employing both mirroring and sequential selection is significantly better than the algorithm without sequentialism on five functions in 20D with expected running times that are about 20% smaller.

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

1. INTRODUCTION

Evolution Strategies (ESs) are robust stochastic search algorithms for numerical optimization where the function to be minimized, f, maps the continuous search space $\mathbb{R}^D$ into $\mathbb{R}$ where D is the dimension of the search space. Recently, a new derandomization technique replacing the independent sampling of new solutions by mirrored sampling has been introduced to enhance the performances of ESs [1]. In this paper, we assess quantitatively the improvements that can be brought when using mirrored samples instead of independent ones. To do so, we compare on the BBOB-2010 noiseless testbed the (1,4)-Covariance-Matrix-Adaptation Evolution-Strategy (CMA-ES) with two variants: first the $(1,4_{\mathrm{m}})$-CMA-ES where mirrored samples are used, and second the $(1,4^{\mathrm{s}}_{\mathrm{m}})$-CMA-ES that in addition to the mirrored samples uses the concept of sequential selection [1]. Both variants are described in Sec. 2.

2. THE ALGORITHMS TESTED

The three algorithms tested are variants of the well-known CMA-ES [10, 9, 8] where at each iteration n, λ new solutions are generated by sampling *independently* λ random vectors $(\mathcal{N}_i(\mathbf{0}, \boldsymbol{C}_n))_{1 \leq i \leq \lambda}$ following a multivariate normal distribution with mean vector $\mathbf{0}$ and covariance matrix $\boldsymbol{C}_n$. The vectors are added to the current solution $\boldsymbol{X}_n$ to create the λ new solutions or offspring $\boldsymbol{X}_n^i = \boldsymbol{X}_n + \sigma_n \mathcal{N}_i(\mathbf{0}, \boldsymbol{C}_n)$ where σ_n is a strictly positive parameter called step-size. In the simple (1,4)-CMA-ES, the number of offspring λ equals 4 and $\boldsymbol{X}_{n+1}$ is set to the best solution among $\boldsymbol{X}_n^1, \ldots, \boldsymbol{X}_n^4$, i.e., $\boldsymbol{X}_{n+1} = \mathrm{argmin}\{f(\boldsymbol{X}_n^1), \ldots, f(\boldsymbol{X}_n^4)\}$.

In the mirrored variant, denoted $(1,4_{\mathrm{m}})$-CMA-ES, the second and fourth offspring are replaced by the offspring symmetric to the first and third with respect to $\boldsymbol{X}_n$, namely $\boldsymbol{X}_n^2 = \boldsymbol{X}_n - \sigma_n \mathcal{N}_1(\mathbf{0}, \boldsymbol{C}_n)$ and $\boldsymbol{X}_n^4 = \boldsymbol{X}_n - \sigma_n \mathcal{N}_3(\mathbf{0}, \boldsymbol{C}_n)$, where $\sigma_n \mathcal{N}_1(\mathbf{0}, \boldsymbol{C}_n)$ and $\sigma_n \mathcal{N}_3(\mathbf{0}, \boldsymbol{C}_n)$ are the random vectors added to $\boldsymbol{X}_n$ to get $\boldsymbol{X}_n^1$ and $\boldsymbol{X}_n^3$. The first/second and third/fourth offspring are, thus, negatively correlated. The update of $\boldsymbol{X}_{n+1}$ is then identical to the (1,4)-CMA-ES, namely $\boldsymbol{X}_{n+1} = \mathrm{argmin}\{f(\boldsymbol{X}_n^1), \ldots, f(\boldsymbol{X}_n^4)\}$.

In the $(1,4^{\mathrm{s}}_{\mathrm{m}})$-CMA-ES, sequential selection is implemented. The four offspring solutions are generated with mirrored sampling. Evaluations are carried out in a sequential manner: after evaluating solution $\boldsymbol{X}_n^i$, it is compared to $\boldsymbol{X}_n$ and if $f(\boldsymbol{X}_n^i) \leq f(\boldsymbol{X}_n)$, the sequence of evaluations is stopped and $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n^i$. In case all four offspring are worse than $\boldsymbol{X}_n$, $\boldsymbol{X}_{n+1} = \mathrm{argmin}\{f(\boldsymbol{X}_n^1), \ldots, f(\boldsymbol{X}_n^4)\}$ according to the comma selection. In sequential selection, the number of offspring evaluated is a random variable by itself ranging from 1 to $\lambda = 4$—allowing to reduce the number of offspring adaptively as long as improvements are easy to achieve [1].

Covariance matrix and step-size are updated using the selected steps [9, 1].

Independent Restarts.

Similar to [3], we independently restarted all algorithms as long as function evaluations were left, where maximally $10^4 \cdot D$ function evaluations have been used.

Parameter setting.

We used the default parameter and termination settings (cf. [1, 5, 8]) found in the source code on the WWW[1] with two exceptions. We rectified the learning rate of the rank-one update of the covariance matrix for small values of λ, setting $c_1 = \min(2, \lambda/3)/((D+1.3)^2 + \mu_{\text{eff}})$. The original value was not designed to work for $\lambda < 5$. We modified the damping parameter for the step-size to $d_\sigma = 0.3{+}2\mu_{\text{eff}}/\lambda{+}c_\sigma$. The setting was found by performing experiments on the sphere function, f_1: d_σ was set as large as possible while still showing close to optimal performance, but, at least as large such that decreasing it by a factor of two did not lead to inacceptable performance. For $\mu_{\text{eff}}/\lambda = 0.35$ and $\mu_{\text{eff}} \leq D + 2$ the former setting of d_σ is recovered. For a smaller ratio of μ_{eff}/λ or for $\mu_{\text{eff}} > D + 2$, the new setting allows larger (i.e. faster) changes of σ. Here, $\mu_{\text{eff}} = 1$. For $\lambda \geq 3$, the new setting might be harmful in a noisy or too rugged landscape. Finally, the step-size multiplier was clamped from above at $\exp(1)$, while we do not believe this had any effect in the presented experiments. Each initial solution $\boldsymbol{X}_0$ was uniformly sampled in $[-4, 4]^D$ and the step-size σ_0 was initialized to 2. The source code used for the experiments is available at[2].

As the same parameter setting has been used for all test functions, the crafting effort CrE of all algorithms is 0.

3. CPU TIMING EXPERIMENTS

For the timing experiment, all three algorithms were run on f_8 with a maximum of $10^4 D$ function evaluations and restarted until at least 30 seconds have passed (according to Figure 2 in [6]). The experiments have been conducted with an 8 core Intel Xeon E5520 machine with 2.27 GHz under Ubuntu 9.1 linux and Matlab R2008a. The time per function evaluation was 3.3; 3.3; 3.0; 3.1; 3.4; 4.0 times 10^{-4} seconds for (1,4)-CMA-ES, 3.1; 3.0; 3.0; 3.2; 3.4; 4.0 times 10^{-4} seconds for (1,4_{m})-CMA-ES, and 7.1; 7.3; 7.7; 8.1; 7.1; 8.1 times 10^{-4} seconds for (1,4^{s}_{m})-CMA-ES in dimensions 2; 3; 5; 10; 20; 40 respectively. Note that MATLAB distributes the computations over all 8 cores only for 20D and 40D.

4. RESULTS

In this section, experiments according to [6] on the benchmark functions given in [4, 7] are presented. The **expected running time (ERT)**, used in the figures and table, depends on a given target function value, $f_{\text{t}} = f_{\text{opt}} + \Delta f$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not reach f_{t}, summed over all trials and divided by the number of trials that actually reached f_{t} [6, 11]. **Statistical significance** is tested with the rank-sum test for a given target Δf_{t} using, for each trial, either the number of needed function evaluations to reach Δf_{t} (inverted and multiplied by -1), or, if the target was not reached, the best Δf-value achieved, measured only up to the smallest number of overall function evaluations for any unsuccessful trial under consideration.

4.1 Comparing (1,4$_{\text{m}}$)- With (1,4)-CMA-ES

The (1,4_{m})-CMA-ES is compared with the (1,4)-CMA-ES in Figures 1 and 2 and in Table 1. Mirroring within the (1,4_{m})-CMA-ES seems to have an important positive impact compared to the baseline (1,4)-CMA-ES. In 20D, we observe a slight worsening only on the Gallagher functions f_{21} and f_{22} which are not statistically significant. On the other hand, the (1,4_{m})-CMA-ES outperforms the (1,4)-CMA-ES on 10 functions of which 4 show statistically significant differences (in 20D and for a target of 10^{-7}): on the sphere function (f_1), the expected running time is 15% lower, on the separable ellipsoid (f_2) it is 18% lower, on the non-separable ellipsoid (f_{10}) 19% lower and on the attractive sector function (f_6) the expected running times even differ by a factor of about 3. These differences are less significant in 5D (Table 1), but Fig. 1 shows similar improvements also for 10D and smaller dimensions.

4.2 Comparing (1,4$^{\text{s}}_{\text{m}}$)- With (1,4$_{\text{m}}$)-CMA-ES

Results comparing the (1,4_{m})-CMA-ES and the (1,4^{s}_{m})-CMA-ES are presented in Figures 3 and 4 and in Table 2.

The results show that the (1,4^{s}_{m})-CMA-ES is statistically significantly better than the (1,4_{m})-CMA-ES on five functions in 20D with expected running times that are about 26% smaller on f_1 and f_2, 25% smaller on f_{10}, 20% smaller on f_{11}, and about 20% smaller on f_{14} than the ones for the (1,4_{m})-CMA-ES. Contrary, there are only two functions where the (1,4^{s}_{m})-CMA-ES is worse than the (1,4_{m})-CMA-ES (13% on f_6 and about 70% on f_{22}) but these differences are not statistically significant. To conclude, the (1,4^{s}_{m})-CMA-ES should be clearly preferred over the (1,4_{m})-CMA-ES on the BBOB-2010 testbed and, according to Sec. 4.1, also over the (1,4)-CMA-ES. Worth to mention is also the fact, that the (1,4^{s}_{m})-CMA-ES shows better performance than the best algorithm from the BBOB-2009 benchmarking on the attractive sector function (f_6) in 5D by about 30% and a slightly better performance on the ellipsoid (f_{10}) in 20D, see [2].

4.3 Comparing (1,4$^{\text{s}}_{\text{m}}$)- With (1,4)-CMA-ES

Regarding this comparison, we refrain from showing the plots and tables due to space limitations. Not surprisingly when considering Sec. 4.2, the results show a statistically significant improvement for the (1,4^{s}_{m})-CMA-ES over the (1,4)-CMA-ES on 7 functions in 20D: f_1 (37% faster), f_2 (39% faster), f_5 (faster by a factor of about 2), f_6 (about 3 times faster), f_{10} (39% faster), f_{11} (25% faster), and f_{14} (about 30% faster). Only on the Gallagher function with 21 peaks (f_{22}), an increase of the expected running time by a factor of about 2 can be observed which is, however, not statistically significant due to the low number of successful runs.

5. CONCLUSIONS

The idea behind derandomization by means of mirroring introduced in [1] is to use only one random sample from a multivariate normal distribution to create two (negatively correlated or *mirrored*) offspring. Thereby, one offspring is generated by adding a random sample to the parent solution

[1] `cmaes.m`, version `3.41.beta`, from `http://www.lri.fr/~hansen/cmaes_inmatlab.html`

[2] `http://coco.gforge.inria.fr/doku.php?id=bbob-2010-results`

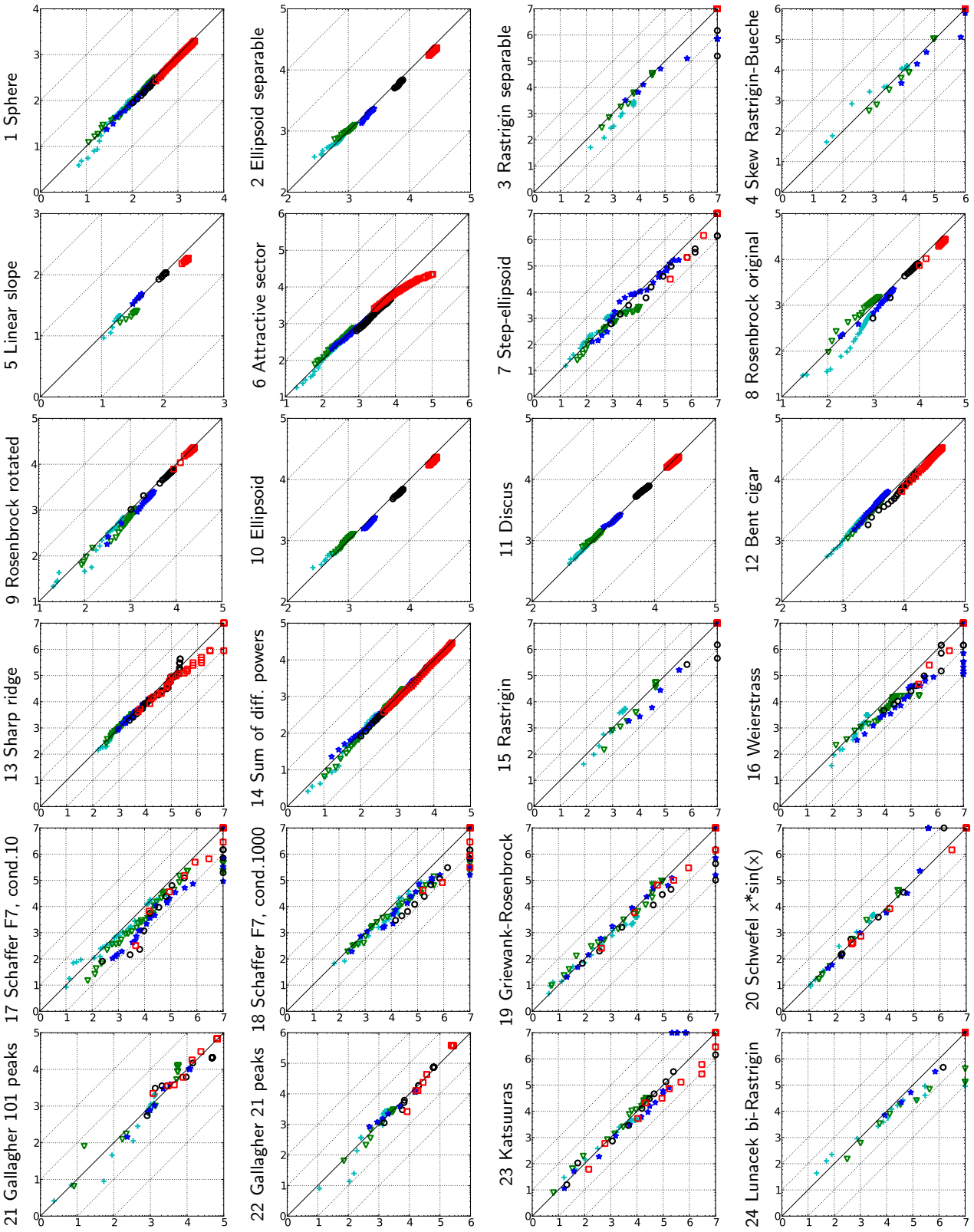

Figure 1: Expected running time (ERT in log10 of number of function evaluations) of (1,4$_{\mathrm{m}}$)-CMA-ES versus (1,4)-CMA-ES for 46 target values $\Delta f \in [10^{-8}, 10]$ in each dimension for functions f_1–f_{24}. Markers on the upper or right edge indicate that the target value was never reached by (1,4$_{\mathrm{m}}$)-CMA-ES or (1,4)-CMA-ES respectively. Markers represent dimension: 2:+, 3:▽, 5:⋆, 10:○, 20:□.

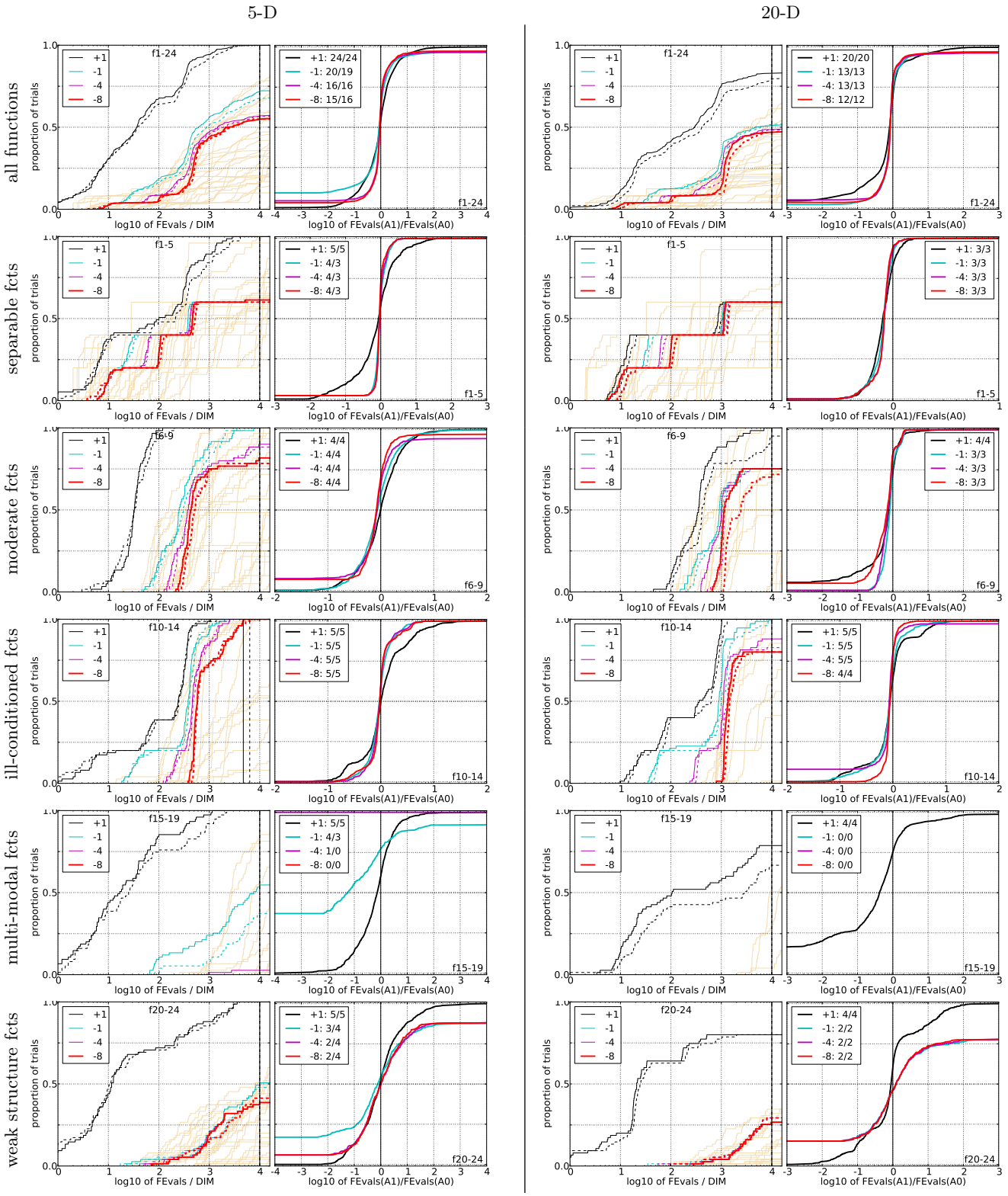

Figure 2: Empirical cumulative distributions (ECDF) of run lengths and speed-up ratios in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of function evaluations divided by dimension D (FEvals/D) to reach a target value $f_{\mathrm{opt}} + \Delta f$ with $\Delta f = 10^k$, where $k \in \{1, -1, -4, -8\}$ is given by the first value in the legend, for (1,4$_{\mathrm{m}}$)-CMA-ES (solid) and (1,4)-CMA-ES (dashed). Light beige lines show the ECDF of FEvals for target value $\Delta f = 10^{-8}$ of algorithms benchmarked during BBOB-2009. Right sub-columns: ECDF of FEval ratios of (1,4$_{\mathrm{m}}$)-CMA-ES divided by (1,4)-CMA-ES, all trial pairs for each function. Pairs where both trials failed are disregarded, pairs where one trial failed are visible in the limits being > 0 or < 1. The legends indicate the number of functions that were solved in at least one trial ((1,4$_{\mathrm{m}}$)-CMA-ES first).

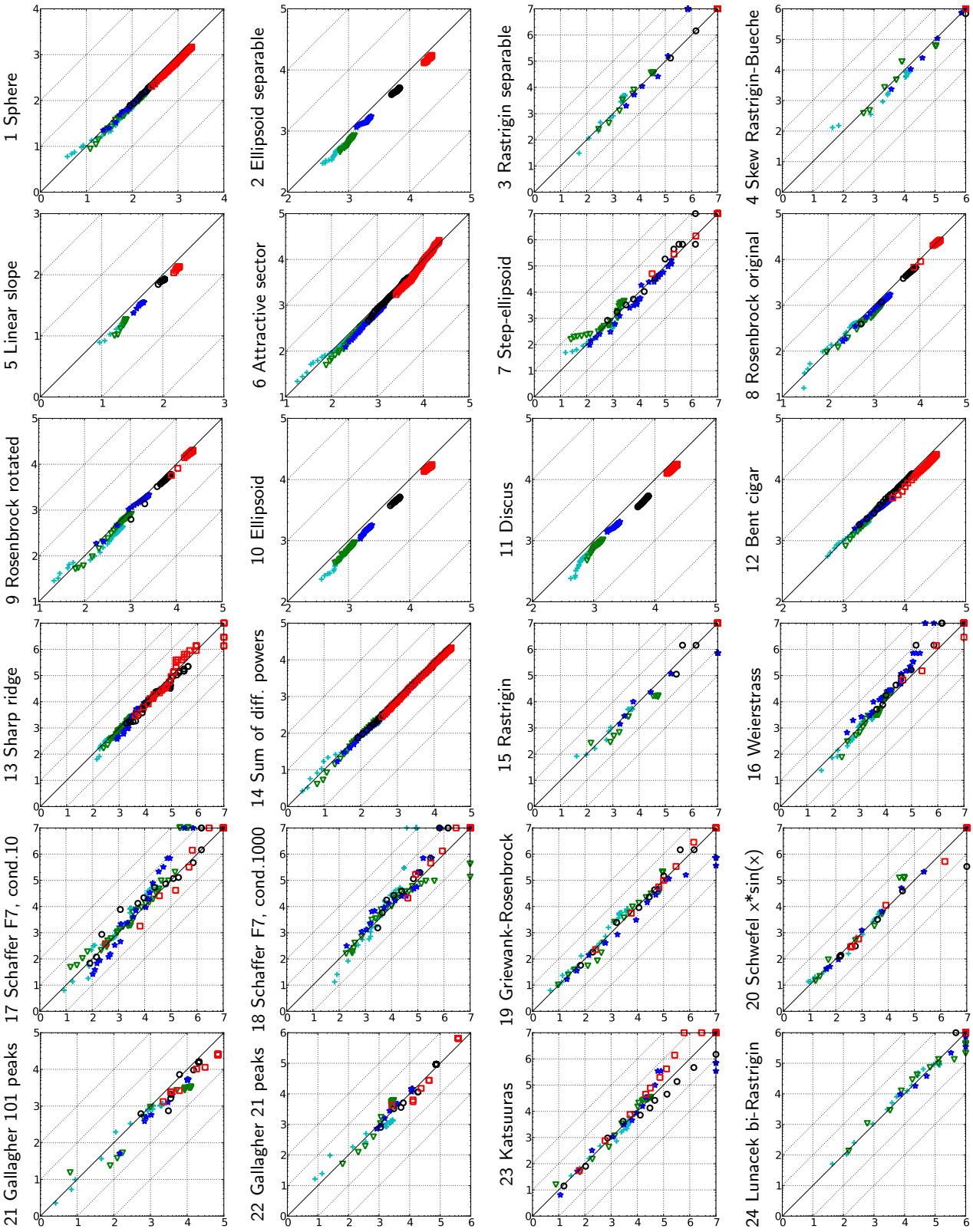

Figure 3: Expected running time (ERT in log10 of number of function evaluations) of $(1,4^{s}_{m})$-CMA-ES versus $(1,4_{m})$-CMA-ES for 46 target values $\Delta f \in [10^{-8}, 10]$ in each dimension for functions f_1–f_{24}. Markers on the upper or right edge indicate that the target value was never reached by $(1,4^{s}_{m})$-CMA-ES or $(1,4_{m})$-CMA-ES respectively. Markers represent dimension: 2:+, 3:▽, 5:⋆, 10:○, 20:□.

5-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f_1	11	12	12	12	12	12	15/15
(1,4)-CMA-ES	2.5	6.9	11	21	30	40	15/15
(1,4$_m$)-CMA-ES	2.2	6.2	10	19	27	37	15/15
f_2	83	87	88	90	92	94	15/15
(1,4)-CMA-ES	20	22	24	25	26	27	15/15
(1,4$_m$)-CMA-ES	16	19	22	23	24	**24***	15/15
f_3	720	1600	1600	1600	1700	1700	15/15
(1,4)-CMA-ES	4.3430		∞	∞	∞	∞*5.0e4*	0/15
(1,4$_m$)-CMA-ES	4.5	79	440	440	440	440	1/15
f_4	810	1600	1700	1800	1900	1900	15/15
(1,4)-CMA-ES	10	∞	∞	∞	∞	∞*5.0e4*	0/15
(1,4$_m$)-CMA-ES	4.6450		∞	∞	∞	∞*5.0e4*	0/15
f_5	10	10	10	10	10	10	15/15
(1,4)-CMA-ES	3.2	4.3	4.5	4.5	4.5	4.5	15/15
(1,4$_m$)-CMA-ES	3.3	4.5	4.8	4.8	4.8	4.8	15/15
f_6	110	210	280	580	1000	1300	15/15
(1,4)-CMA-ES	1.6	1.8	1.8	1.5	1.3	1.2	15/15
(1,4$_m$)-CMA-ES	1.8	1.6	1.7	1.3	1	**0.96***	15/15
f_7	24	320	1200	1600	1600	1600	15/15
(1,4)-CMA-ES	7.1	3.1	5.8	41	41	97	2/15
(1,4$_m$)-CMA-ES	5.5	3.7	7.1	35	35	80	4/15
f_8	73	270	340	390	410	420	15/15
(1,4)-CMA-ES	2.6	4.9	5.7	5.9	6.2	6.3	15/15
(1,4$_m$)-CMA-ES	2.9	3.6	4.3	4.8	5	5.2	15/15
f_9	35	130	210	300	340	370	15/15
(1,4)-CMA-ES	9	14	11	9.4	9	8.6	15/15
(1,4$_m$)-CMA-ES	5.2	9.5	8.2	7.2	7	6.7	15/15
f_{10}	350	500	570	630	830	880	15/15
(1,4)-CMA-ES	5	4.1	3.9	3.9	3.1	3	15/15
(1,4$_m$)-CMA-ES	4.5	3.6	3.3	3.4	2.7	2.7	15/15
f_{11}	140	200	760	1200	1500	1700	15/15
(1,4)-CMA-ES	10	9.9	2.9	2	1.7	1.6	15/15
(1,4$_m$)-CMA-ES	12	9.8	2.8	2	1.7	1.6	15/15
f_{12}	110	270	370	460	1300	1500	15/15
(1,4)-CMA-ES	14	9	8.9	8.6	3.6	3.5	15/15
(1,4$_m$)-CMA-ES	14	10	10	10	4.2	4.1	15/15
f_{13}	130	190	250	1300	1800	2300	15/15
(1,4)-CMA-ES	7	9.7	11	3	4.2	5.1	15/15
(1,4$_m$)-CMA-ES	6.4	11	9.8	2.8	3.2	3.9	15/15
f_{14}	9.8	41	58	140	250	480	15/15
(1,4)-CMA-ES	1.7	1.9	3.1	4.2	6.8	5.4	15/15
(1,4$_m$)-CMA-ES	2.3	2.1	2.9	3.9	6	5.3	15/15
f_{15}	510	9300	1.9e4	2.0e4	2.1e4	2.1e4	14/15
(1,4)-CMA-ES	8.2	38	∞	∞	∞	∞*5.0e4*	0/15
(1,4$_m$)-CMA-ES	3.7	18	∞	∞	∞	∞*5.0e4*	0/15
f_{16}	120	610	2700	1.0e4	1.2e4	1.2e4	15/15
(1,4)-CMA-ES	7.2	27	31	∞	∞	∞*5.0e4*	0/15
(1,4$_m$)-CMA-ES	2.9	**5.8***	12	11	∞	∞*5.0e4*	0/15
f_{17}	5.2	210	900	3700	6400	7900	15/15
(1,4)-CMA-ES	120	20	18	∞	∞	∞*5.0e4*	0/15
(1,4$_m$)-CMA-ES	20	2.3	4.1	90	∞	∞*5.0e4*	0/15
f_{18}	100	380	4000	9300	1.1e4	1.2e4	15/15
(1,4)-CMA-ES	3	31	21	∞	∞	∞*5.0e4*	0/15
(1,4$_m$)-CMA-ES	1.9	5.9	9.2	∞	∞	∞*5.0e4*	0/15
f_{19}	1	1	240	1.2e5	1.2e5	1.2e5	15/15
(1,4)-CMA-ES	21	6.5e3	∞	∞	∞	∞*5.0e4*	0/15
(1,4$_m$)-CMA-ES	20	9.0e3	660	∞	∞	∞*5.0e4*	0/15
f_{20}	16	850	3.8e4	5.4e4	5.5e4	5.5e4	14/15
(1,4)-CMA-ES	3.3	10	9.4	6.6	6.6	6.5	2/15
(1,4$_m$)-CMA-ES	2.8	6.9	∞	∞	∞	∞*5.0e4*	0/15
f_{21}	41	1200	1700	1700	1700	1800	14/15
(1,4)-CMA-ES	5.7	2.9	6.9	6.8	6.8	6.7	14/15
(1,4$_m$)-CMA-ES	3.5	3.4	6.1	6.1	6	6	15/15
f_{22}	71	390	940	1000	1000	1100	14/15
(1,4)-CMA-ES	6.9	13	16	15	15	15	14/15
(1,4$_m$)-CMA-ES	12	10	12	12	12	11	14/15
f_{23}	3	520	1.4e4	3.2e4	3.3e4	3.4e4	15/15
(1,4)-CMA-ES	5.7	28	7	11	22	21	1/15
(1,4$_m$)-CMA-ES	3.8	12	4.2	∞	∞	∞*5.0e4*	0/15
f_{24}	1600	2.2e5	6.4e6	9.6e6	1.3e7	1.3e7	3/15
(1,4)-CMA-ES	5.3	∞	∞	∞	∞	∞*5.0e4*	0/15
(1,4$_m$)-CMA-ES	4.4	∞	∞	∞	∞	∞*5.0e4*	0/15

20-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f_1	43	43	43	43	43	43	15/15
(1,4)-CMA-ES	7.7	13	18	27	38	49	15/15
(1,4$_m$)-CMA-ES	6.2	11	15	**24**$^{\star 2}$	**32**$^{\star 3}$	**42**$^{\star 3}$	15/15
f_2	380	390	390	390	390	390	15/15
(1,4)-CMA-ES	54	62	65	68	69	70	15/15
(1,4$_m$)-CMA-ES	**45***	**51**$^{\star 2}$	**54**$^{\star 3}$	**56**$^{\star 3}$	**57**$^{\star 3}$	**58**$^{\star 3}$	15/15
f_3	5100	7600	7600	7600	7600	7700	15/15
(1,4)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
f_4	4700	7600	7700	7700	7800	1.4e5	9/15
(1,4)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
f_5	41	41	41	41	41	41	15/15
(1,4)-CMA-ES	5.1	6.1	6.4	6.5	6.5	6.5	15/15
(1,4$_m$)-CMA-ES	3.7	4.4	4.5	4.6	4.6	4.6	15/15
f_6	1300	2300	3400	5200	6700	8400	15/15
(1,4)-CMA-ES	2.1	1.9	2	2.5	3.6	7.4	13/15
(1,4$_m$)-CMA-ES	2	1.6	1.6	**1.5**$^{\star 2}$	**1.8**$^{\star 2}$	**2.2**$^{\star 3}$	15/15
f_7	1400	4300	9500	1.7e4	1.7e4	1.7e4	15/15
(1,4)-CMA-ES	120	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	23	∞	∞	∞	∞	∞*2.0e5*	0/15
f_8	2000	3900	4000	4200	4400	4500	15/15
(1,4)-CMA-ES	4.9	7.7	8	8.1	8	8	15/15
(1,4$_m$)-CMA-ES	3.6	5.8	6	6.1	6.1	6.1	15/15
f_9	1700	3100	3300	3500	3600	3700	15/15
(1,4)-CMA-ES	5.1	6	6.5	6.6	6.6	6.6	15/15
(1,4$_m$)-CMA-ES	4.6	5.7	6.1	6.2	6.2	6.1	15/15
f_{10}	7400	8700	1.1e4	1.5e4	1.7e4	1.7e4	15/15
(1,4)-CMA-ES	2.8	2.8	2.4	1.8	1.6	1.6	15/15
(1,4$_m$)-CMA-ES	2.3	**2.2**$^{\star 2}$	**1.9**$^{\star 2}$	**1.5**$^{\star 3}$	**1.3**$^{\star 3}$	**1.3**$^{\star 3}$	15/15
f_{11}	1000	2200	6300	9800	1.2e4	1.5e4	15/15
(1,4)-CMA-ES	16	8	3.1	2.2	1.9	1.6	15/15
(1,4$_m$)-CMA-ES	16	7.8	2.9	2.1	1.8	1.5	15/15
f_{12}	1000	1900	2700	4100	1.2e4	1.4e4	15/15
(1,4)-CMA-ES	8.8	8.8	8.5	7.2	2.9	2.9	15/15
(1,4$_m$)-CMA-ES	6.1	6.7	6.5	5.7	2.3	2.3	15/15
f_{13}	650	2000	2800	1.9e4	2.4e4	3.0e4	15/15
(1,4)-CMA-ES	8.3	15	25	21	120	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	6.4	9.8	16	8	36	∞*2.0e5*	0/15
f_{14}	75	240	300	930	1600	1.6e4	15/15
(1,4)-CMA-ES	5.7	3.1	3.6	3.9	7.3	1.6	15/15
(1,4$_m$)-CMA-ES	5	2.8	3.1	**3.2***	6.4	1.4	15/15
f_{15}	3.0e4	1.5e5	3.1e5	3.2e5	4.5e5	4.6e5	15/15
(1,4)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
f_{16}	1400	2.7e4	7.7e4	1.9e5	2.0e5	2.2e5	15/15
(1,4)-CMA-ES	140	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	34	∞	∞	∞	∞	∞*2.0e5*	0/15
f_{17}	63	1000	4000	3.1e4	5.6e4	8.0e4	15/15
(1,4)-CMA-ES	73	2.8e3	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	5.1640		∞	∞	∞	∞*2.0e5*	0/15
f_{18}	620	4000	2.0e4	6.8e4	1.3e5	1.5e5	15/15
(1,4)-CMA-ES	260	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	67	∞	∞	∞	∞	∞*2.0e5*	0/15
f_{19}	1	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
(1,4)-CMA-ES	440	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	260	1.4e6	∞	∞	∞	∞*2.0e5*	0/15
f_{20}	82	4.6e4	3.1e6	5.5e6	5.6e6	5.6e6	14/15
(1,4)-CMA-ES	5.1	61	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	4.6	31	∞	∞	∞	∞*2.0e5*	0/15
f_{21}	560	6500	1.4e4	1.5e4	1.6e4	1.8e4	15/15
(1,4)-CMA-ES	2.1	3.6	4.7	4.5	4.3	3.8	14/15
(1,4$_m$)-CMA-ES	3.9	4.7	4.8	4.7	4.4	3.9	14/15
f_{22}	470	5600	2.3e4	2.5e4	2.7e4	1.3e5	12/15
(1,4)-CMA-ES	18	6.6	11	11	10	2	8/15
(1,4$_m$)-CMA-ES	5.6	7.7	16	15	14	2.9	6/15
f_{23}	3.2	1600	6.7e4	4.9e5	8.1e5	8.4e5	15/15
(1,4)-CMA-ES	45	110	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	19	45	∞	∞	∞	∞*2.0e5*	0/15
f_{24}	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
(1,4)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4$_m$)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15

Table 1: Expected running time (ERT in number of function evaluations) divided by the best ERT measured during BBOB-2009 (given in the respective first row) for the algorithms (1,4)-CMA-ES and is (1,4$_m$)-CMA-ES for different Δf values for functions f_1–f_{24}. The median number of conducted function evaluations is additionally given in *italics*, if $\mathrm{ERT}(10^{-7}) = \infty$. #succ is the number of trials that reached the final target $f_{\mathrm{opt}} + 10^{-8}$. Bold entries are statistically significantly better compared to the other algorithm, with $p = 0.05$ or $p = 10^{-k}$ where $k > 1$ is the number following the $\star$ symbol, with Bonferroni correction of 48.

and a second offspring then equals the solution which is symmetric to the first one with respect to the parent (by adding the negative sample to the parent). Here, this concept of mirroring has been integrated within two variants of a simple (1,4)-CMA-ES (of which the (1,4$^{\mathrm{s}}_{\mathrm{m}}$)-CMA-ES uses sequential selection [1] in addition and the (1,4$_m$)-CMA-ES does not) and compared on the noiseless BBOB-2010 testbed.

The results show that using mirroring gives an improvement of at least 15% on 4 functions (stat. significant in 20D and 10^{-7} as target). Combining mirroring with sequential selection is even better—showing further improvements over the (1,4$_m$)-CMA-ES by at least 20% on 5 functions.

Acknowledgments This work was supported by the French national research agency (ANR) in the projects SYSCOMM ANR-08-SYSC-017 and COSINUS ANR-08-COSI-007-12.

6. REFERENCES

[1] A. Auger, D. Brockhoff, and N. Hansen. Mirrored sampling and sequential selection for evolution strategies. Rapport de Recherche RR-7249, INRIA Saclay—Île-de-France, April 2010.

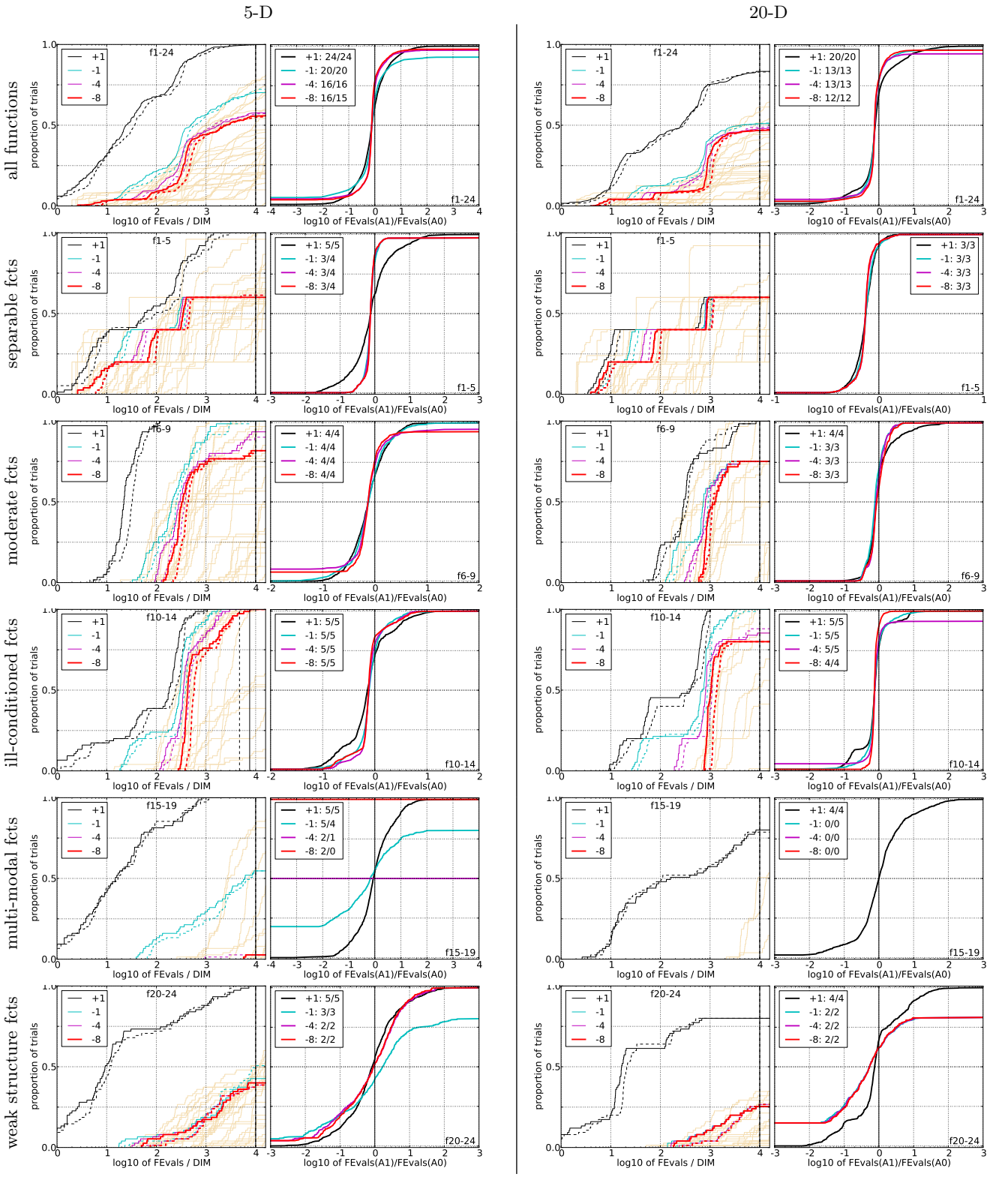

Figure 4: Empirical cumulative distributions of run lengths of (1,4$^{\mathrm{s}}_{\mathrm{m}}$)-CMA-ES (solid) and (1,4$_{\mathrm{m}}$)-CMA-ES (dashed) and speed-up ratios of the former divided by the latter in 5-D (left) and 20-D (right) as in Fig. 2.

[2] 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.

[3] A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 1777–1784, 2005.

[4] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter

5-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f_1	11	12	12	12	12	12	15/15
(1,4$_m$)-CMA-ES	2.2	6.2	10	19	27	37	15/15
(1,4^s_m)-CMA-ES	2	5	8.3	15	**22***	**30***	15/15
f_2	83	87	88	90	92	94	15/15
(1,4$_m$)-CMA-ES	16	19	22	23	24	24	15/15
(1,4^s_m)-CMA-ES	14	15	**16**$^{\star 3}$	**17**$^{\star 3}$	**18**$^{\star 3}$	**18**$^{\star 3}$	15/15
f_3	720	1600	1600	1600	1700	1700	15/15
(1,4$_m$)-CMA-ES	4.5	79	440	440	440	440	1/15
(1,4^s_m)-CMA-ES	2.8	98	∞	∞	∞	∞*5.0e4*	0/15
f_4	810	1600	1700	1800	1900	1900	15/15
(1,4$_m$)-CMA-ES	4.6	450	∞	∞	∞	∞*5.0e4*	0/15
(1,4^s_m)-CMA-ES	2.9	460	∞	∞	∞	∞*5.0e4*	0/15
f_5	10	10	10	10	10	10	15/15
(1,4$_m$)-CMA-ES	3.3	4.5	4.8	4.8	4.8	4.8	15/15
(1,4^s_m)-CMA-ES	2.4	3.5	3.6	3.6	3.6	3.6	15/15
f_6	110	210	280	580	1000	1300	15/15
(1,4$_m$)-CMA-ES	1.8	1.6	1.7	1.3	1	0.96	15/15
(1,4^s_m)-CMA-ES	1.1	1.1	1.2	**0.91***	**0.7**$^{\star 2}$	**0.69***	15/15
f_7	24	320	1200	1600	1600	1600	15/15
(1,4$_m$)-CMA-ES	5.5	3.7	7.1	35	35	80	4/15
(1,4^s_m)-CMA-ES	4.1	1.9	3.9	25	25	60	4/15
f_8	73	270	340	390	410	420	15/15
(1,4$_m$)-CMA-ES	2.9	3.6	4.3	4.8	5	5.2	15/15
(1,4^s_m)-CMA-ES	2.3	2.9	3.5	3.7	3.8	4	15/15
f_9	35	130	210	300	340	370	15/15
(1,4$_m$)-CMA-ES	5.2	9.5	8.2	7.2	7	6.7	15/15
(1,4^s_m)-CMA-ES	5.4	10	7.7	6.3	6	5.8	15/15
f_{10}	350	500	570	630	830	880	15/15
(1,4$_m$)-CMA-ES	4.5	3.6	3.3	3.4	2.7	2.7	15/15
(1,4^s_m)-CMA-ES	**3.1***	**2.6***	**2.6***	**2.5**$^{\star 3}$	**2**$^{\star 3}$	**2**$^{\star 3}$	15/15
f_{11}	140	200	760	1200	1500	1700	15/15
(1,4$_m$)-CMA-ES	12	9.8	2.8	2	1.7	1.6	15/15
(1,4^s_m)-CMA-ES	9.8	**7.9***	**2.2**$^{\star 2}$	**1.5**$^{\star 3}$	**1.3**$^{\star 3}$	**1.2**$^{\star 3}$	15/15
f_{12}	110	270	370	460	1300	1500	15/15
(1,4$_m$)-CMA-ES	14	10	10	10	4.2	4.1	15/15
(1,4^s_m)-CMA-ES	15	9.5	8.9	8.6	3.5	3.3	15/15
f_{13}	130	190	250	1300	1800	2300	15/15
(1,4$_m$)-CMA-ES	6.4	11	9.8	2.8	3.2	3.9	15/15
(1,4^s_m)-CMA-ES	2.9	4.9	7.7	2.5	3.1	3.1	15/15
f_{14}	9.8	41	58	140	250	480	15/15
(1,4$_m$)-CMA-ES	2.3	2.1	2.9	3.9	6	5.3	15/15
(1,4^s_m)-CMA-ES	1.7	1.8	2.3	2.9	**4.5**$^{\star 2}$	**4**$^{\star 3}$	15/15
f_{15}	510	9300	1.9e4	2.0e4	2.1e4	2.1e4	14/15
(1,4$_m$)-CMA-ES	3.7	18	∞	∞	∞	∞*5.0e4*	0/15
(1,4^s_m)-CMA-ES	2.8	13	38	36	35	34	1/15
f_{16}	120	610	2700	1.0e4	1.2e4	1.2e4	15/15
(1,4$_m$)-CMA-ES	2.9	5.8	12	11	∞	∞*5.0e4*	0/15
(1,4^s_m)-CMA-ES	5.6	16	19	33	∞	∞*5.0e4*	0/15
f_{17}	5.2	210	900	3700	6400	7900	15/15
(1,4$_m$)-CMA-ES	20	2.3	4.1	90	∞	∞*5.0e4*	0/15
(1,4^s_m)-CMA-ES	5.1	0.71	4.5	∞	∞	∞*5.0e4*	0/15
f_{18}	100	380	4000	9300	1.1e4	1.2e4	15/15
(1,4$_m$)-CMA-ES	1.9	5.9	9.2	∞	∞	∞*5.0e4*	0/15
(1,4^s_m)-CMA-ES	3.1	18	6.4	∞	∞	∞*5.0e4*	0/15
f_{19}	1	1	240	1.2e5	1.2e5	1.2e5	15/15
(1,4$_m$)-CMA-ES	20	9.0e3	660	∞	∞	∞*5.0e4*	0/15
(1,4^s_m)-CMA-ES	17	3.1e3	480	6.1	6	6	1/15
f_{20}	16	850	3.8e4	5.4e4	5.5e4	5.5e4	14/15
(1,4$_m$)-CMA-ES	2.8	6.9	∞	∞	∞	∞*5.0e4*	0/15
(1,4^s_m)-CMA-ES	2.6	7.6	∞	∞	∞	∞*5.0e4*	0/15
f_{21}	41	1200	1700	1700	1700	1800	14/15
(1,4$_m$)-CMA-ES	3.5	3.4	6.1	6.1	6	6	15/15
(1,4^s_m)-CMA-ES	1.2	1.7	3.1	3.1	3.1	3.1	15/15
f_{22}	71	390	940	1000	1000	1100	14/15
(1,4$_m$)-CMA-ES	12	10	12	12	12	11	14/15
(1,4^s_m)-CMA-ES	10	12	15	14	14	14	15/15
f_{23}	3	520	1.4e4	3.2e4	3.3e4	3.4e4	15/15
(1,4$_m$)-CMA-ES	3.8	12	4.2	∞	∞	∞*5.0e4*	0/15
(1,4^s_m)-CMA-ES	2.2	8.9	24	∞	∞	∞*5.0e4*	0/15
f_{24}	1600	2.2e5	6.4e6	9.6e6	1.3e7	1.3e7	3/15
(1,4$_m$)-CMA-ES	4.4	∞	∞	∞	∞	∞*5.0e4*	0/15
(1,4^s_m)-CMA-ES	6	3.3	∞	∞	∞	∞*5.0e4*	0/15

20-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f_1	43	43	43	43	43	43	15/15
(1,4$_m$)-CMA-ES	6.2	11	15	24	32	42	15/15
(1,4^s_m)-CMA-ES	**4.7**$^{\star 2}$	**8.1**$^{\star 3}$	**11**$^{\star 3}$	**18**$^{\star 3}$	**24**$^{\star 3}$	**31**$^{\star 3}$	15/15
f_2	380	390	390	390	390	390	15/15
(1,4$_m$)-CMA-ES	45	51	54	56	57	58	15/15
(1,4^s_m)-CMA-ES	**34**$^{\star 3}$	**39**$^{\star 3}$	**41**$^{\star 3}$	**42**$^{\star 3}$	**43**$^{\star 3}$	**43**$^{\star 3}$	15/15
f_3	5100	7600	7600	7600	7600	7700	15/15
(1,4$_m$)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
f_4	4700	7600	7700	7700	7800	1.4e5	9/15
(1,4$_m$)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
f_5	41	41	41	41	41	41	15/15
(1,4$_m$)-CMA-ES	3.7	4.4	4.5	4.6	4.6	4.6	15/15
(1,4^s_m)-CMA-ES	**2.7***	3.2	3.3	3.3	3.3	3.3	15/15
f_6	1300	2300	3400	5200	6700	8400	15/15
(1,4$_m$)-CMA-ES	2	1.6	1.6	1.5	1.8	2.2	15/15
(1,4^s_m)-CMA-ES	**1.3***	**1.1***	**1**$^{\star 2}$	1.2	1.7	2.5	15/15
f_7	1400	4300	9500	1.7e4	1.7e4	1.7e4	15/15
(1,4$_m$)-CMA-ES	23	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	37	∞	∞	∞	∞	∞*2.0e5*	0/15
f_8	2000	3900	4000	4200	4400	4500	15/15
(1,4$_m$)-CMA-ES	3.6	5.8	6	6.1	6.1	6.1	15/15
(1,4^s_m)-CMA-ES	3.3	5.8	5.9	5.9	5.8	5.8	15/15
f_9	1700	3100	3300	3500	3600	3700	15/15
(1,4$_m$)-CMA-ES	4.6	5.7	6.1	6.2	6.2	6.1	15/15
(1,4^s_m)-CMA-ES	**3.3***	5.2	5.4	5.4	5.4	5.3	15/15
f_{10}	7400	8700	1.1e4	1.5e4	1.7e4	1.7e4	15/15
(1,4$_m$)-CMA-ES	2.3	2.2	1.9	1.5	1.3	1.3	15/15
(1,4^s_m)-CMA-ES	**1.8**$^{\star 2}$	**1.7**$^{\star 2}$	**1.5**$^{\star 3}$	**1.1**$^{\star 3}$	**0.98**$^{\star 3}$	**0.98**$^{\star 3}$	15/15
f_{11}	1000	2200	6300	9800	1.2e4	1.5e4	15/15
(1,4$_m$)-CMA-ES	16	7.8	2.9	2.1	1.8	1.5	15/15
(1,4^s_m)-CMA-ES	13	6.1	**2.3***	**1.6***	**1.4***	**1.2***	15/15
f_{12}	1000	1900	2700	4100	1.2e4	1.4e4	15/15
(1,4$_m$)-CMA-ES	6.1	6.7	6.5	5.7	2.3	2.3	15/15
(1,4^s_m)-CMA-ES	4.8	5.1	5	4.4	1.8	1.8	15/15
f_{13}	650	2000	2800	1.9e4	2.4e4	3.0e4	15/15
(1,4$_m$)-CMA-ES	6.4	9.8	16	8	36	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	4.5	6.6	11	21	56	∞*2.0e5*	0/15
f_{14}	75	240	300	930	1600	1.6e4	15/15
(1,4$_m$)-CMA-ES	5	2.8	3.1	3.2	6.4	1.4	15/15
(1,4^s_m)-CMA-ES	3.8	2.1	**2.3***	**2.6***	**5**$^{\star 3}$	**1.1**$^{\star 3}$	15/15
f_{15}	3.0e4	1.5e5	3.1e5	3.2e5	4.5e5	4.6e5	15/15
(1,4$_m$)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
f_{16}	1400	2.7e4	7.7e4	1.9e5	2.0e5	2.2e5	15/15
(1,4$_m$)-CMA-ES	34	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	49	∞	∞	∞	∞	∞*2.0e5*	0/15
f_{17}	63	1000	4000	3.1e4	5.6e4	8.0e4	15/15
(1,4$_m$)-CMA-ES	5.1	640	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	6.1	1.4e3	∞	∞	∞	∞*2.0e5*	0/15
f_{18}	620	4000	2.0e4	6.8e4	1.3e5	1.5e5	15/15
(1,4$_m$)-CMA-ES	67	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	34	∞	∞	∞	∞	∞*2.0e5*	0/15
f_{19}	1	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
(1,4$_m$)-CMA-ES	260	1.4e6	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	230	2.8e6	∞	∞	∞	∞*2.0e5*	0/15
f_{20}	82	4.6e4	3.1e6	5.5e6	5.6e6	5.6e6	14/15
(1,4$_m$)-CMA-ES	4.6	31	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	**3.5***	11	∞	∞	∞	∞*2.0e5*	0/15
f_{21}	560	6500	1.4e4	1.5e4	1.6e4	1.8e4	15/15
(1,4$_m$)-CMA-ES	3.9	4.7	4.8	4.7	4.4	3.9	14/15
(1,4^s_m)-CMA-ES	2.4	1.7	1.8	1.7	1.6	1.5	15/15
f_{22}	470	5600	2.3e4	2.5e4	2.7e4	1.3e5	12/15
(1,4$_m$)-CMA-ES	5.6	7.7	16	15	14	2.9	6/15
(1,4^s_m)-CMA-ES	9	5	28	26	24	4.9	4/15
f_{23}	3.2	1600	6.7e4	4.9e5	8.1e5	8.4e5	15/15
(1,4$_m$)-CMA-ES	19	45	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	17	120	∞	∞	∞	∞*2.0e5*	0/15
f_{24}	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
(1,4$_m$)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15
(1,4^s_m)-CMA-ES	∞	∞	∞	∞	∞	∞*2.0e5*	0/15

Table 2: Expected running time (ERT in number of function evaluations) divided by the best ERT measured during BBOB-2009 as in Table 1 but now comparing (1,4$_m$)-CMA-ES and (1,4^s_m)-CMA-ES.

black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.

[5] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In F. Rothlauf, editor, *GECCO (Companion)*, pages 2389–2396. ACM, 2009.

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

[7] 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. Updated February 2010.

[8] 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.

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

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

[11] K. Price. Differential evolution vs. the functions of the second ICEO. In *Proceedings of the IEEE International Congress on Evolutionary Computation*, pages 153–157, 1997.