Investigating the Impact of Sequential Selection in the (1,4)-CMA-ES 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

This paper investigates the impact of sequential selection, a concept recently introduced for Evolution Strategies (ESs). Sequential selection performs the evaluations of the different candidate solutions sequentially and concludes the iteration immediately if one offspring is better than the parent. In this paper, the (1,4^s)-CMA-ES, where sequential selection is implemented, is compared on the BBOB-2010 noiseless testbed to the (1,4)-CMA-ES. For each strategy, an independent restart mechanism is implemented. A total budget of $10^4 D$ function evaluations per trial has been used, where D is the dimension of the search space.

The experiments show for the (1,4^s)-CMA-ES a statistically significant worsening compared to the (1,4)-CMA-ES only on the attractive sector function but a significant improvement by about 20% on 5 out of the 24 BBOB-2010 functions (sphere, separable and rotated ellipsoid, discus, and sum of different powers).

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 robust stochastic search algorithms for numerical optimization where the function to be minimized, f, maps the continuous search space $\mathbb{R}^D$ into $\mathbb{R}$. In ESs, a population of λ candidate solutions is sampled at each iteration by adding to a current solution λ random vectors following a multivariate normal distribution. In the local search $(1, \lambda)$-ES we are interested in, the best of the λ solutions, i.e., the solution having the smallest objective function value, is selected to become the new current solution. Recently, a new selection called *sequential selection* has been introduced to enhance the performance of $(1, \lambda)$-ESs [1]. Sequential selection consists in performing the λ offspring-evaluations sequentially and concluding the iteration as soon as one offspring is better than the parent.

In this paper, we assess quantitatively the possible gain that can be brought by sequential selection. To this end, we have implemented sequential selection within the well-known Covariance-Matrix-Adaptation Evolution-Strategy (CMA-ES) [9, 8, 7]. We compare on the BBOB-2010 testbed the performance of the (1,4^s)-CMA-ES implementing sequential selection with the performance of the (1,4)-CMA-ES.

2. THE ALGORITHMS TESTED

The algorithms tested are derived from the standard CMA-ES algorithm where at each iteration n, λ new solutions, or offspring, 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 $\boldsymbol{X}_n^i = \boldsymbol{X}_n + \sigma_n \mathcal{N}_i(\mathbf{0}, \boldsymbol{C}_n)$ where σ_n is a strictly positive parameter called step-size [8].

We benchmark two variants of the CMA-ES algorithm where λ equals 4, namely the (1,4)-CMA-ES and the (1,4^s)-CMA-ES. Both algorithms differ in the way $\boldsymbol{X}_{n+1}$ is updated:

1. in the (1,4)-CMA-ES, $\boldsymbol{X}_{n+1}$ is the best among the four offspring, i.e., $\boldsymbol{X}_{n+1} = \text{argmin}\{f(\boldsymbol{X}_n^1), \ldots, f(\boldsymbol{X}_n^4)\}$,
2. in the (1,4^s)-CMA-ES, $\boldsymbol{X}_n^1$ is first evaluated and compared to $\boldsymbol{X}_n$, if $f(\boldsymbol{X}_n^1) \leq f(\boldsymbol{X}_n)$, then $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n^1$, else $\boldsymbol{X}_n^2$ is evaluated and compared to $\boldsymbol{X}_n$, if $f(\boldsymbol{X}_n^2) \leq f(\boldsymbol{X}_n)$, then $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n^2$ else $\boldsymbol{X}_n^3$ is evaluated ... else $\boldsymbol{X}_n^4$ is evaluated and the best among the four offspring is chosen, i.e., $\boldsymbol{X}_{n+1} = \text{argmin}\{f(\boldsymbol{X}_n^1), \ldots, f(\boldsymbol{X}_n^4)\}$.

Note that when sequential selection is applied, the number of offspring evaluated is a random variable, ranging here from 1 to $\lambda = 4$.

Covariance matrix and step-size are updated using the selected steps [8, 1].

2.1 Independent Restarts

Similar to [2], we independently restarted the (1,4)-CMA-ES and the $(1,4^s)$-CMA-ES as long as function evaluations were left, where $10^4 \cdot D$ has been used as the maximal number of function evaluations.

2.2 Parameter Settings

We used the default parameter and termination settings (cf. [1, 4, 7]) 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 in all experiments for all test functions, the crafting effort CrE of all three 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 [5]). 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 and 7.7; 7.4; 7.5; 7.9; 7.3; 8.1 times 10^{-4} seconds for $(1,4^s)$-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. COMPARING THE (1,4) AND THE $(1,4^s)$-CMA-ES

Results from experiments comparing (1,4)-CMA-ES and $(1,4^s)$-CMA-ES according to [5] on the benchmark functions given in [3, 6] are presented in Figures 1, 2 and 3 and in Table 1. The **expected running time (ERT)**, used in the figures and table, depends on a given target function value, $f_t = f_{\text{opt}} + \Delta f$, 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 [5, 10]. **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.

From Fig. 2 and 3 as well as from Table 1, we observe that the expected running time of the $(1,4^s)$-CMA-ES is by approximately 20% smaller than the one of the (1,4)-CMA-ES on the sphere f_1, the separable (f_2) and rotated (f_{10}) ellipsoid, the discus f_{11}, and the sum of different powers function (f_{14}, all results statistically significant). Moreover, only on the attractive sector function (f_6), the $(1,4^s)$-CMA-ES shows a statistically significant worse performance than the (1,4)-CMA-ES.

For the Gallagher functions (f_{21} and f_{22}), mixed results can be observed: on f_{21}, the success probability of the $(1,4^s)$-CMA-ES is slightly higher than the one of the (1,4)-CMA-ES whereas on f_{22}, the success probability is lower, resulting in an expected running time that is more than twice as large as for the (1,4)-CMA-ES (both results are not statistically significant).

5. CONCLUSIONS

The idea behind the sequential selection scheme introduced in [1] is to skip function evaluations of the λ offspring in a $(1 \overset{+}{,} \lambda)$-ES as soon as an offspring is evaluated which is better than the current solution. Here, the concept of sequential selection has been integrated into a comma-strategy, the so-called $(1,4^s)$-CMA-ES, and compared with the baseline algorithm (1,4)-CMA-ES on the BBOB-2010 testbed.

The experiments show improved results for the algorithm employing sequential selection: the $(1,4^s)$-CMA-ES shows a significant improvement over the (1,4)-CMA-ES by about 20% on 5 of the 24 BBOB-2010 functions. However, a statistically significant worsening on the attractive sector function in comparison to the (1,4)-CMA-ES is reported as well. Also on the Gallagher function f_{22}, the $(1,4^s)$-CMA-ES shows a lower success probability than the (1,4)-CMA-ES but the difference is not statistically significant here.

6. ACKNOWLEDGMENTS

This work received 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.

7. 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.

[2] 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.

[3] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking

[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`

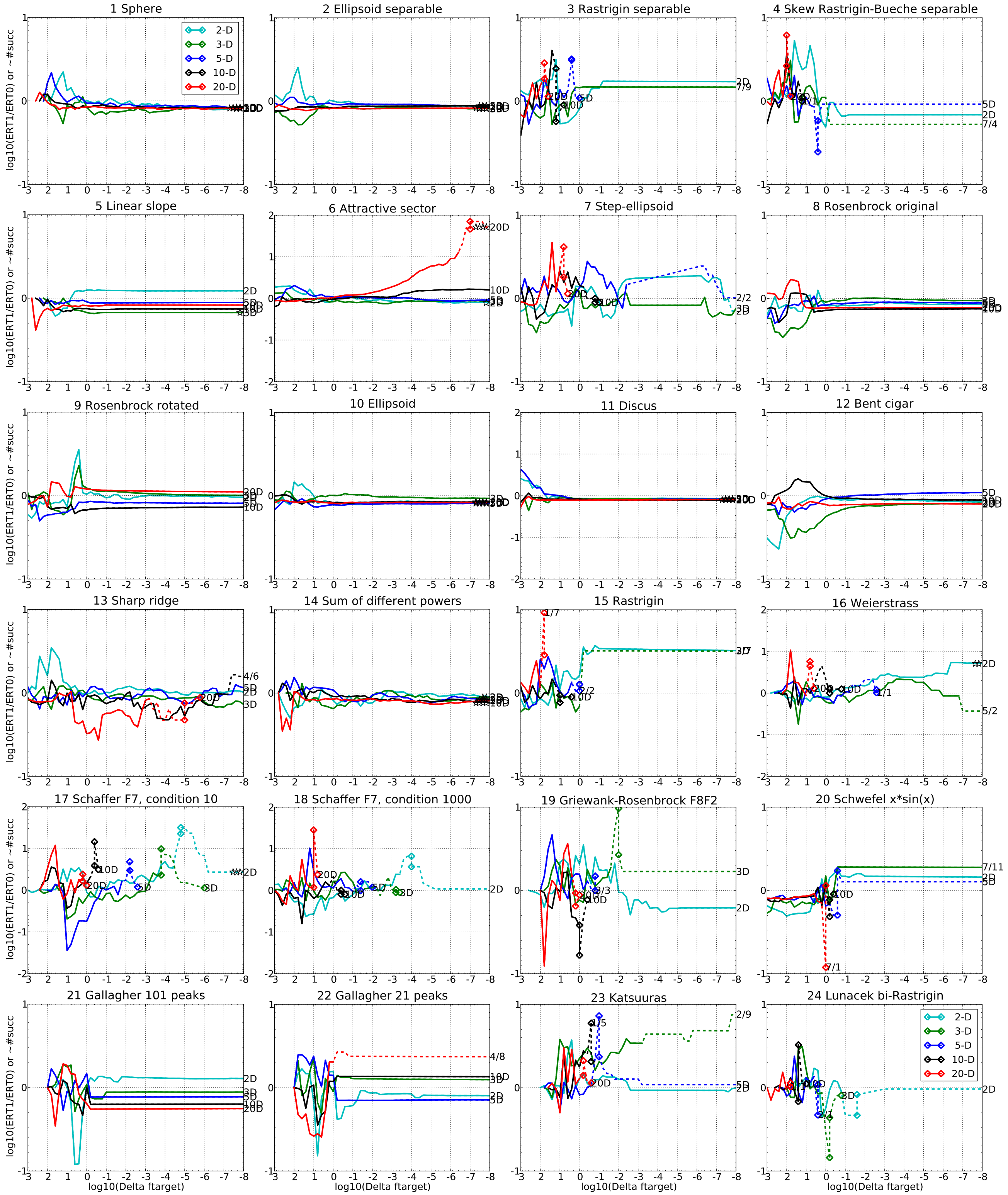

Figure 1: ERT ratio of (1,4^s)-CMA-ES divided by (1,4)-CMA-ES versus $\log_{10}(\Delta f)$ for f_1–f_{24} in 2, 3, 5, 10, 20. Ratios $< 10^0$ indicate an advantage of (1,4^s)-CMA-ES, smaller values are always better. The line gets dashed when for any algorithm the ERT exceeds thrice the median of the trial-wise overall number of f-evaluations for the same algorithm on this function. Symbols indicate the best achieved Δf-value of one algorithm (ERT gets undefined to the right). The dashed line continues as the fraction of successful trials of the other algorithm, where 0 means 0% and the y-axis limits mean 100%, values below zero for (1,4^s)-CMA-ES. The line ends when no algorithm reaches Δf anymore. The number of successful trials is given, only if it was in $\{1 \ldots 9\}$ for (1,4^s)-CMA-ES (1st number) and non-zero for (1,4)-CMA-ES (2nd number). Results are significant with $p = 0.05$ for one star and $p = 10^{-\#\star}$ otherwise, with Bonferroni correction within each figure.

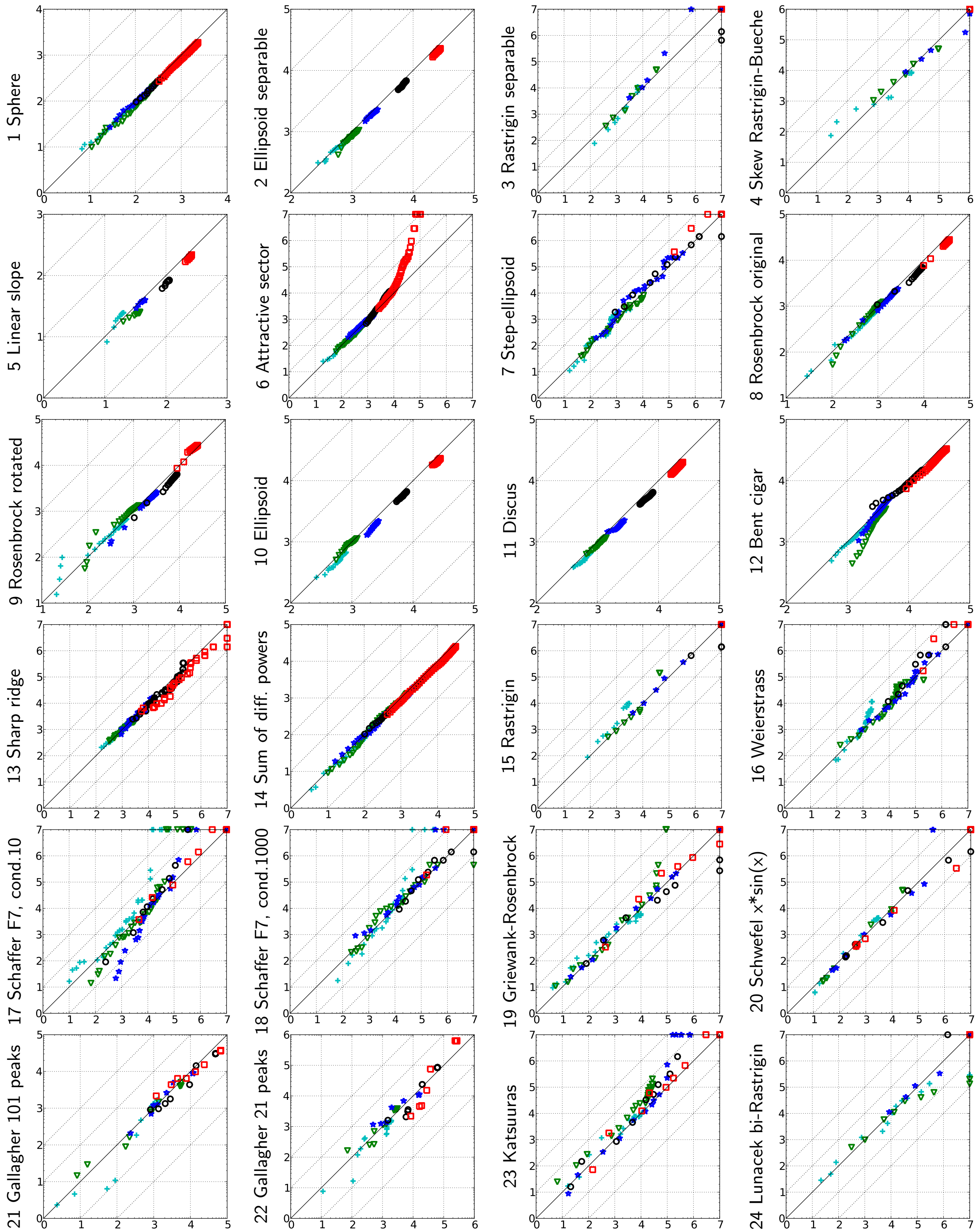

Figure 2: Expected running time (ERT in log10 of number of function evaluations) of (1,4^s)-CMA-ES versus (1,4)-CMA-ES for 46 target values $\Delta f \in [10^{-8}, 10]$ in each dimension for functions f_1–f_{24}. Markers on the upper or right edge indicate that the target value was never reached by (1,4^s)-CMA-ES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

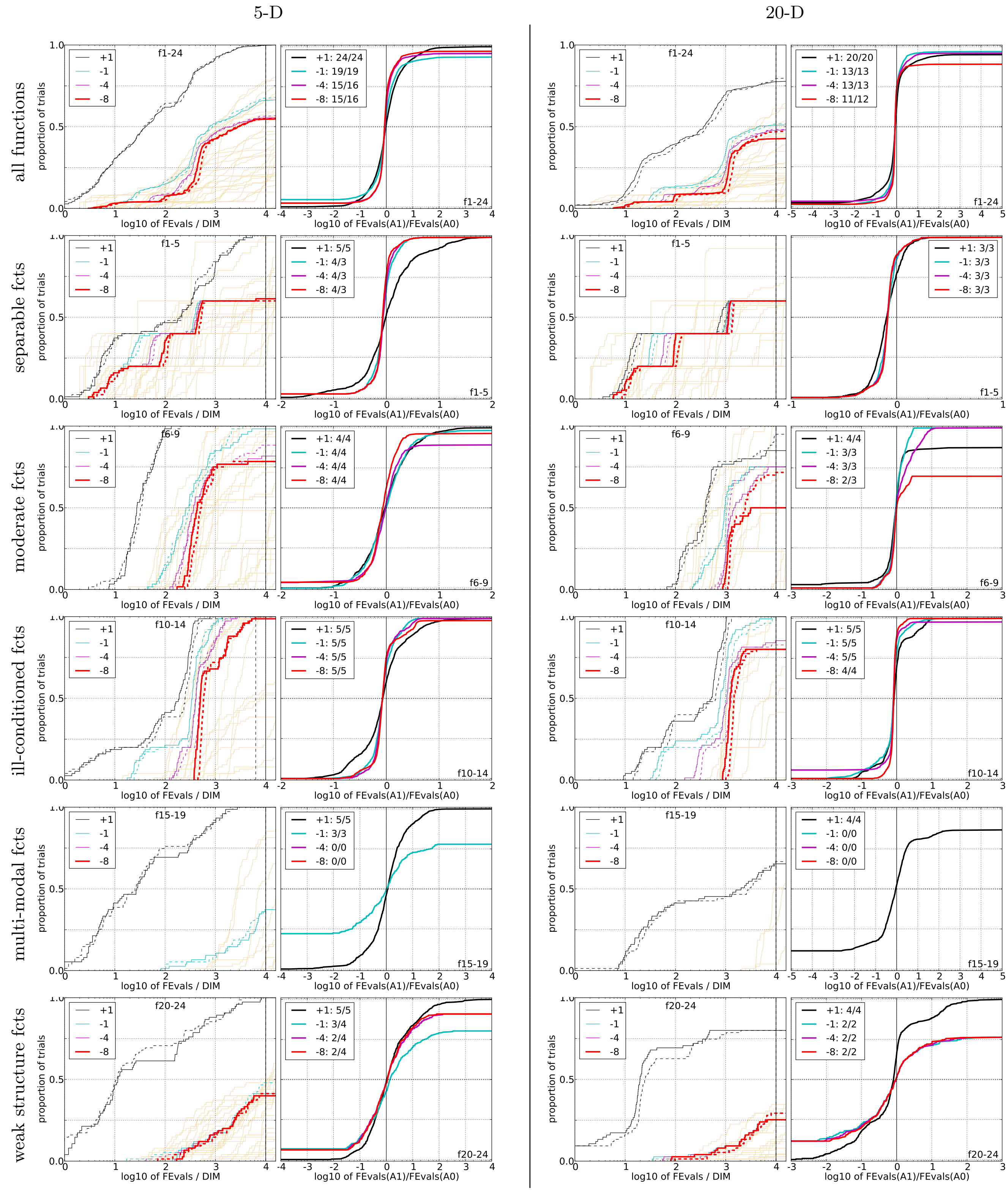

Figure 3: Empirical cumulative distributions (ECDF) of run lengths and speed-up ratios in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of function evaluations divided by dimension D (FEvals/D) to reach a target value $f_{\mathrm{opt}} + \Delta f$ with $\Delta f = 10^k$, where $k \in \{1, -1, -4, -8\}$ is given by the first value in the legend, for (1,4$^{\mathrm{s}}$)-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$^{\mathrm{s}}$)-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$^{\mathrm{s}}$)-CMA-ES first).

5-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
$\mathbf{f_1}$	11	12	12	12	12	12	15/15
(1,4)-CMA-ES	2.5	6.9	11	21	30	40	15/15
(1,4^s)-CMA-ES	2.5	6.5	10	18	26	34	15/15
$\mathbf{f_2}$	83	87	88	90	92	94	15/15
(1,4)-CMA-ES	20	22	24	25	26	27	15/15
(1,4^s)-CMA-ES	18	21	22	23	23	24	15/15
$\mathbf{f_3}$	720	1600	1600	1600	1700	1700	15/15
(1,4)-CMA-ES	4.3	430	∞	∞	∞	∞ *5.0e4*	0/15
(1,4^s)-CMA-ES	5.9	∞	∞	∞	∞	∞ *5.0e4*	0/15
$\mathbf{f_4}$	810	1600	1700	1800	1900	1900	15/15
(1,4)-CMA-ES	10	∞	∞	∞	∞	∞ *5.0e4*	0/15
(1,4^s)-CMA-ES	11	450	430	400	390	380	1/15
$\mathbf{f_5}$	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^s)-CMA-ES	2.9	3.9	4	4	4	4	15/15
$\mathbf{f_6}$	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^s)-CMA-ES	1.8	1.9	1.9	1.5	1.1	1.1	15/15
$\mathbf{f_7}$	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^s)-CMA-ES	8.4	4.2	11	100	100	140	2/15
$\mathbf{f_8}$	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^s)-CMA-ES	2.4	4.1	5	5.4	5.5	5.6	15/15
$\mathbf{f_9}$	35	130	210	300	340	370	15/15
(1,4)-CMA-ES	9	14	11	9.4	9	8.6	15/15
(1,4^s)-CMA-ES	5.7	11	9.2	7.7	7.3	6.9	15/15
$\mathbf{f_{10}}$	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^s)-CMA-ES	3.7	3.2	**3.1***	**3.1**$^{\star 2}$	**2.5**$^{\star 3}$	**2.5**$^{\star 3}$	15/15
$\mathbf{f_{11}}$	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^s)-CMA-ES	10	8.1	**2.3***	**1.7***	**1.4***	**1.3**$^{\star 2}$	15/15
$\mathbf{f_{12}}$	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^s)-CMA-ES	9.7	8.2	8.7	9	3.9	3.8	15/15
$\mathbf{f_{13}}$	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^s)-CMA-ES	5	9.7	9.9	3.3	3.1	5	14/15
$\mathbf{f_{14}}$	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^s)-CMA-ES	2	2	2.5	3.8	**5.2**$^{\star 2}$	4.7	15/15
$\mathbf{f_{15}}$	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^s)-CMA-ES	8.5	40	∞	∞	∞	∞ *5.0e4*	0/15
$\mathbf{f_{16}}$	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^s)-CMA-ES	8.1	20	33	∞	∞	∞ *5.0e4*	0/15
$\mathbf{f_{17}}$	5.2	210	900	3700	6400	7900	15/15
(1,4)-CMA-ES	120	20	18	∞	∞	∞ *5.0e4*	0/15
(1,4^s)-CMA-ES	4.2	3.6	18	∞	∞	∞ *5.0e4*	0/15
$\mathbf{f_{18}}$	100	380	4000	9300	1.1e4	1.2e4	15/15
(1,4)-CMA-ES	3	31	21	∞	∞	∞ *5.0e4*	0/15
(1,4^s)-CMA-ES	8.8	35	20	∞	∞	∞ *5.0e4*	0/15
$\mathbf{f_{19}}$	1	1	240	1.2e5	1.2e5	1.2e5	15/15
(1,4)-CMA-ES	21	6.5e3	∞	∞	∞	∞ *5.0e4*	0/15
(1,4^s)-CMA-ES	25	1.0e4	∞	∞	∞	∞ *5.0e4*	0/15
$\mathbf{f_{20}}$	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^s)-CMA-ES	2.8	6.8	∞	∞	∞	∞ *5.0e4*	0/15
$\mathbf{f_{21}}$	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^s)-CMA-ES	5.1	4.3	5.3	5.3	5.2	5.2	15/15
$\mathbf{f_{22}}$	71	390	940	1000	1000	1100	14/15
(1,4)-CMA-ES	6.9	13	16	15	15	15	14/15
(1,4^s)-CMA-ES	16	18	12	11	11	11	15/15
$\mathbf{f_{23}}$	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^s)-CMA-ES	3	23	51	∞	∞	∞ *5.0e4*	0/15
$\mathbf{f_{24}}$	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^s)-CMA-ES	7	∞	∞	∞	∞	∞ *5.0e4*	0/15

20-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
$\mathbf{f_1}$	43	43	43	43	43	43	15/15
(1,4)-CMA-ES	7.7	13	18	27	38	49	15/15
(1,4^s)-CMA-ES	6.3	11	**15**$^{\star 2}$	**23**$^{\star 2}$	**31**$^{\star 3}$	**40**$^{\star 3}$	15/15
$\mathbf{f_2}$	380	390	390	390	390	390	15/15
(1,4)-CMA-ES	54	62	65	68	69	70	15/15
(1,4^s)-CMA-ES	43	**51**$^{\star 2}$	**53**$^{\star 3}$	**55**$^{\star 3}$	**57**$^{\star 3}$	**58**$^{\star 3}$	15/15
$\mathbf{f_3}$	5100	7600	7600	7600	7600	7700	15/15
(1,4)-CMA-ES	∞	∞	∞	∞	∞	∞ *2.0e5*	0/15
(1,4^s)-CMA-ES	∞	∞	∞	∞	∞	∞ *2.0e5*	0/15
$\mathbf{f_4}$	4700	7600	7700	7700	7800	1.4e5	9/15
(1,4)-CMA-ES	∞	∞	∞	∞	∞	∞ *2.0e5*	0/15
(1,4^s)-CMA-ES	∞	∞	∞	∞	∞	∞ *2.0e5*	0/15
$\mathbf{f_5}$	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^s)-CMA-ES	4.1	5	5.4	5.4	5.4	5.4	15/15
$\mathbf{f_6}$	1300	2300	3400	5200	6700	8400	15/15
(1,4)-CMA-ES	2.1	1.9	2	2.5	**3.6**$^{\star 2}$	**7.4**$^{\star 3}$	13/15
(1,4^s)-CMA-ES	2	2	2.4	4.8	22	340	0/15
$\mathbf{f_7}$	1400	4300	9500	1.7e4	1.7e4	1.7e4	15/15
(1,4)-CMA-ES	120	∞	∞	∞	∞	∞ *2.0e5*	0/15
(1,4^s)-CMA-ES	270	∞	∞	∞	∞	∞ *2.0e5*	0/15
$\mathbf{f_8}$	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^s)-CMA-ES	3.8	5.9	6.2	6.2	6.2	6.2	15/15
$\mathbf{f_9}$	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^s)-CMA-ES	5	7.2	7.5	7.5	7.4	7.3	15/15
$\mathbf{f_{10}}$	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^s)-CMA-ES	2.4	**2.3**$^{\star 2}$	**2**$^{\star 2}$	**1.5**$^{\star 3}$	**1.3**$^{\star 3}$	**1.3**$^{\star 3}$	15/15
$\mathbf{f_{11}}$	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^s)-CMA-ES	**13***	**6.3**$^{\star 2}$	**2.4**$^{\star 2}$	**1.8**$^{\star 2}$	**1.5**$^{\star 2}$	**1.3**$^{\star 2}$	15/15
$\mathbf{f_{12}}$	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^s)-CMA-ES	7.1	6.9	6.7	5.7	2.3	2.3	15/15
$\mathbf{f_{13}}$	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^s)-CMA-ES	7.6	4.9	14	17	57	∞ *2.0e5*	0/15
$\mathbf{f_{14}}$	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^s)-CMA-ES	4.7	2.6	2.9	**3.2**$^{\star 3}$	**5.4**$^{\star 3}$	**1.3**$^{\star 2}$	15/15
$\mathbf{f_{15}}$	3.0e4	1.5e5	3.1e5	3.2e5	4.5e5	4.6e5	15/15
(1,4)-CMA-ES	∞	∞	∞	∞	∞	∞ *2.0e5*	0/15
(1,4^s)-CMA-ES	∞	∞	∞	∞	∞	∞ *2.0e5*	0/15
$\mathbf{f_{16}}$	1400	2.7e4	7.7e4	1.9e5	2.0e5	2.2e5	15/15
(1,4)-CMA-ES	140	∞	∞	∞	∞	∞ *2.0e5*	0/15
(1,4^s)-CMA-ES	120	∞	∞	∞	∞	∞ *2.0e5*	0/15
$\mathbf{f_{17}}$	63	1000	4000	3.1e4	5.6e4	8.0e4	15/15
(1,4)-CMA-ES	73	2.8e3	∞	∞	∞	∞ *2.0e5*	0/15
(1,4^s)-CMA-ES	59	∞	∞	∞	∞	∞ *2.0e5*	0/15
$\mathbf{f_{18}}$	620	4000	2.0e4	6.8e4	1.3e5	1.5e5	15/15
(1,4)-CMA-ES	260	∞	∞	∞	∞	∞ *2.0e5*	0/15
(1,4^s)-CMA-ES	300	∞	∞	∞	∞	∞ *2.0e5*	0/15
$\mathbf{f_{19}}$	1	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
(1,4)-CMA-ES	440	∞	∞	∞	∞	∞ *2.0e5*	0/15
(1,4^s)-CMA-ES	330	2.8e6	∞	∞	∞	∞ *2.0e5*	0/15
$\mathbf{f_{20}}$	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^s)-CMA-ES	4.4	7.3	∞	∞	∞	∞ *2.0e5*	0/15
$\mathbf{f_{21}}$	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^s)-CMA-ES	3.8	2.3	2.6	2.5	2.4	2.1	15/15
$\mathbf{f_{22}}$	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^s)-CMA-ES	4.7	13	27	25	24	4.7	4/15
$\mathbf{f_{23}}$	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^s)-CMA-ES	22	140	∞	∞	∞	∞ *2.0e5*	0/15
$\mathbf{f_{24}}$	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
(1,4)-CMA-ES	∞	∞	∞	∞	∞	∞ *2.0e5*	0/15
(1,4^s)-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 (1,4^s)-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.

2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.