



HAL
open science

Investigating the Impact of Sequential Selection in the (1,2)-CMA-ES on the Noiseless BBOB-2010 Testbed

Anne Auger, Dimo Brockhoff, Nikolaus Hansen

► **To cite this version:**

Anne Auger, Dimo Brockhoff, Nikolaus Hansen. Investigating the Impact of Sequential Selection in the (1,2)-CMA-ES on the Noiseless BBOB-2010 Testbed. GECCO workshop on Black-Box Optimization Benchmarking (BBOB'2010), Jul 2010, Portland, OR, United States. pp.1591-1596, 10.1145/1830761.1830777 . inria-00502431

HAL Id: inria-00502431

<https://inria.hal.science/inria-00502431v1>

Submitted on 14 Jul 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Investigating the Impact of Sequential Selection in the (1,2)-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), that consists in performing the evaluations of the different candidate solutions sequentially, concluding the iteration immediately if one offspring is better than the parent. The performance of the (1,2)-Covariance-Matrix-Adaptation Evolution-Strategy (CMA-ES) is compared to the performance of the (1,2^s)-CMA-ES where sequential selection is implemented on the BBOB-2010 noiseless benchmark testbed. 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 do not allow a general statement regarding a statistically significant difference between the two algorithms and we conclude that the sequential selection has no impact on the performance of the (1,2)-CMA-ES on the noiseless BBOB-2009 testbed.

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, λ)-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, λ)-ESs [2]. 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 want to assess quantitatively the possible gain that can be brought by sequential selection. We have implemented sequential selection within the well-known Covariance-Matrix-Adaptation Evolution-Strategy (CMA-ES) [10, 9, 8]. We compare on the BBOB-2010 testbed the performances of the (1,2)-CMA-ES with the performances of the (1,2^s)-CMA-ES implementing sequential selection.

2. THE ALGORITHMS TESTED

The algorithms tested are derived from the standard CMA-ES algorithm where at each iteration n , λ new solutions are generated by sampling independently λ random vectors $(\mathcal{N}_i(\mathbf{0}, \mathbf{C}_n))_{1 \leq i \leq \lambda}$ following a multivariate normal distribution with mean vector $\mathbf{0}$ and covariance matrix \mathbf{C}_n . The vectors are added to the current solution \mathbf{X}_n to create the λ new solutions or offspring $\mathbf{X}_n^i = \mathbf{X}_n + \sigma_n \mathcal{N}_i(\mathbf{0}, \mathbf{C}_n)$ where σ_n is a strictly positive parameter called step-size [9].

We benchmark two variants of the CMA-ES algorithm where λ equals 2, namely the (1,2)-CMA-ES and the (1,2^s)-CMA-ES. Both algorithms differ in the way \mathbf{X}_{n+1} is updated:

1. in the (1,2)-CMA-ES, \mathbf{X}_{n+1} is the best among \mathbf{X}_n^1 and \mathbf{X}_n^2 , i.e., $\mathbf{X}_{n+1} = \operatorname{argmin}\{f(\mathbf{X}_n^1), f(\mathbf{X}_n^2)\}$,
2. in the (1,2^s)-CMA-ES, \mathbf{X}_n^1 is first compared to \mathbf{X}_n , if $f(\mathbf{X}_n^1) \leq f(\mathbf{X}_n)$, then $\mathbf{X}_{n+1} = \mathbf{X}_n^1$, else $\mathbf{X}_{n+1} = \operatorname{argmin}\{f(\mathbf{X}_n^1), f(\mathbf{X}_n^2)\}$.

Note that the idea behind sequential selection is to save function evaluations in case the first offspring is better than the parent solution, since in this case, the second offspring is not evaluated.

Covariance matrix and step-size are updated using the selected steps [9, 2].

2.1 Independent Restarts

Similar to [3], we independently restarted the (1,2)-CMA-ES and the (1,2^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. [2, 5, 8]) found in the source code on the WWW¹ 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 unacceptable 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 \mathbf{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 at².

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

3. CPU TIMING EXPERIMENTS

For the timing experiment, both 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 5.9; 5.9; 6.2; 6.1; 6.8; 9.1 times 10^{-4} seconds for (1,2)-CMA-ES and 9.5; 8.9; 8.9; 8.9; 9.7; 14 times 10^{-4} seconds for (1,2^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 (1,2) AND THE (1,2^s)-CMA-ES

Results from experiments comparing (1,2)-CMA-ES and (1,2^s)-CMA-ES according to [6] on the benchmark functions given in [4, 7] 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 [6, 11]. **Statistical significance** is tested with

¹[cmaes.m](http://www.lri.fr/~hmaes.m), version 3.41.beta, from <http://www.lri.fr/~hmaes.m>

²<http://coco.gforge.inria.fr/doku.php?id=bbob-2010-results>

the rank-sum test for a given target Δf_t (10^{-8} in Figure 1) 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. 1–3 as well as from Table 1, no general statement regarding a statistically significant difference between the two algorithms can be given. Compared to the (1,2)-CMA-ES, five of the 24 functions have been solved faster by the (1,2^s)-CMA-ES, five have been solved slower, and one has been solved as fast as the (1,2)-CMA-ES (in 20D and for a target value of 10^{-7}), whereas the results are statistically significant only for the separable ellipsoid (f_2) where the (1,2)-CMA-ES is 10% faster than the (1,2^s)-CMA-ES.

5. CONCLUSIONS

The idea behind the sequential selection scheme introduced in [2] is to finish the iteration as soon as an offspring is evaluated which is better than the current solution and thereby save some of the λ function evaluations per iteration in a $(1 \ddagger \lambda)$ -ES. Here, the concept of sequential selection has been integrated into a comma-strategy, the so-called (1,2^s)-CMA-ES, and compared with the corresponding baseline (1,2)-CMA-ES on the BBOB-2010 testbed.

The experiments show a mixed performance of the (1,2^s)-CMA-ES: compared to the (1,2)-CMA-ES, five of the 24 functions have been solved faster, five have been solved slower, and one has been solved as fast as the (1,2)-CMA-ES, whereas the results are statistically significant only for one of the functions. Although no significant impact of the sequential selection within the (1,2)-CMA-ES can be reported here, the impact is larger and positive for the (1,4)-CMA-ES as shown in [1].

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.

6. REFERENCES

- [1] 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.
- [2] 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.
- [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 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,

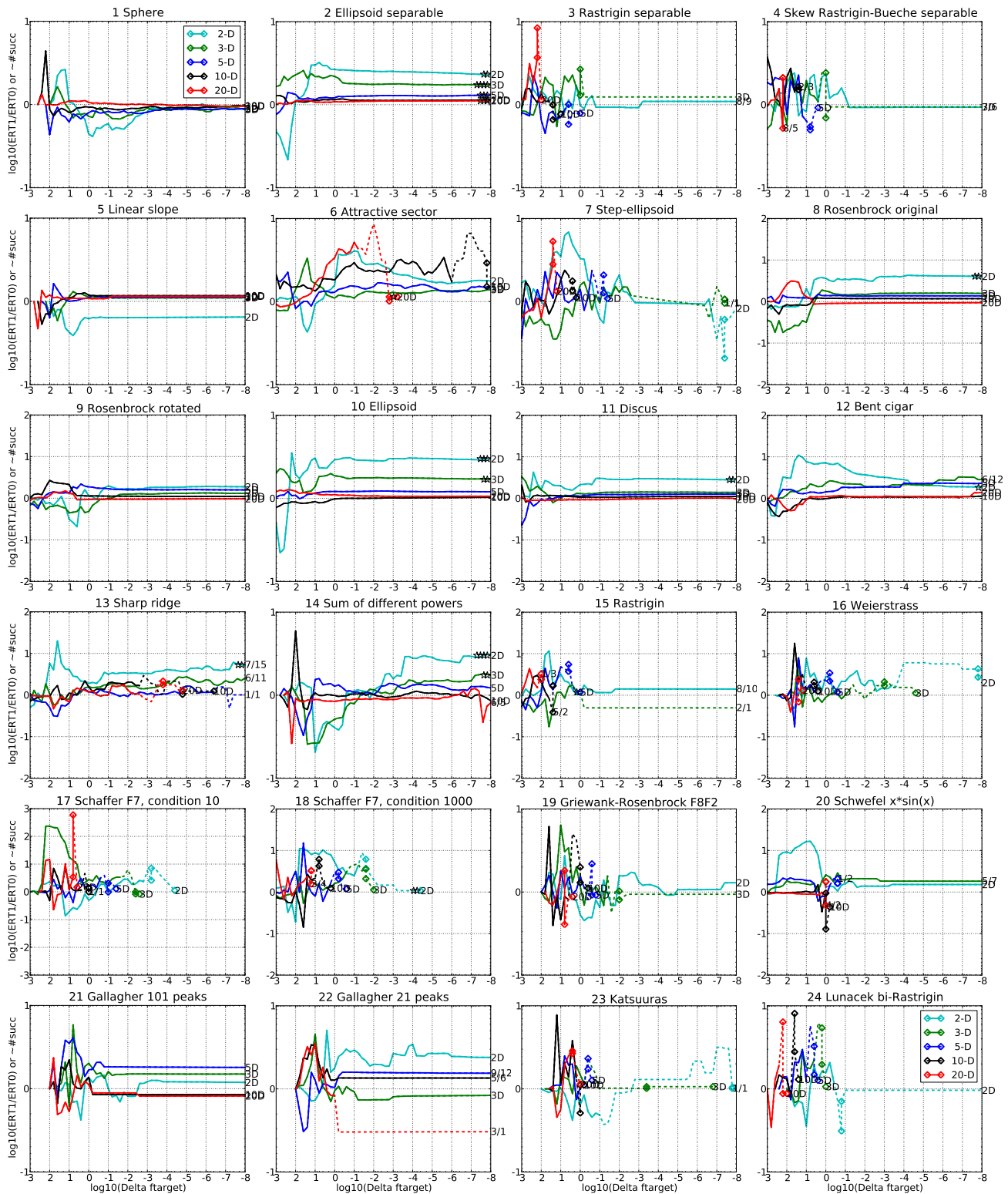


Figure 1: ERT ratio of $(1,2^s)$ -CMA-ES divided by $(1,2)$ -CMA-ES versus $\log_{10}(\Delta f)$ for f_1 - f_{24} in **2**, **3**, **5**, **10**, **20**. Ratios $< 10^0$ indicate an advantage of $(1,2^s)$ -CMA-ES, smaller values are always better. The line gets dashed when for any algorithm the ERT exceeds 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 $(1,2^s)$ -CMA-ES. The line ends when no algorithm reaches Δf anymore. The number of successful trials is given, only if it was in $\{1 \dots 9\}$ for $(1,2^s)$ -CMA-ES (1st number) and non-zero for $(1,2)$ -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.

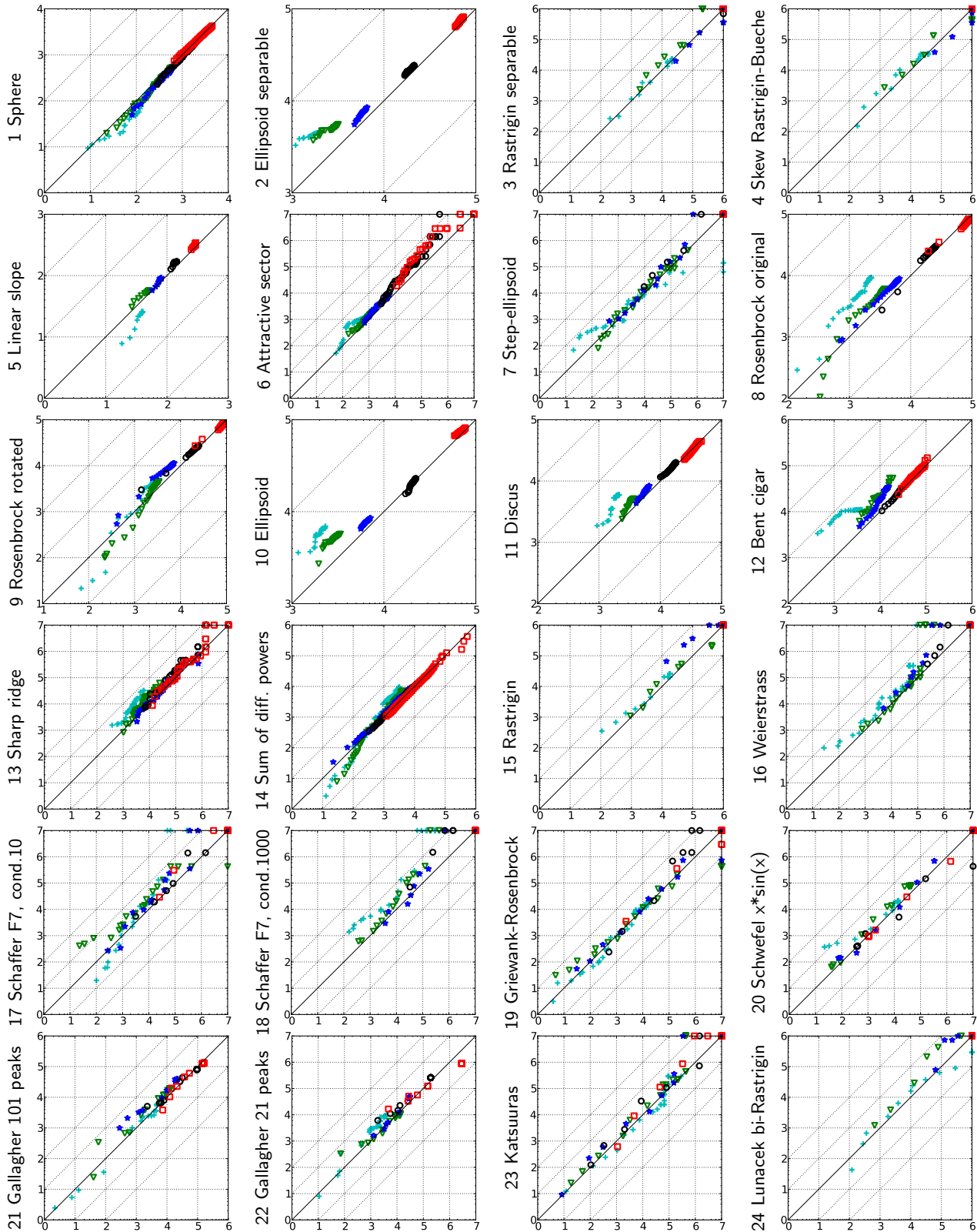


Figure 2: Expected running time (ERT in log10 of number of function evaluations) of $(1,2^s)$ -CMA-ES versus $(1,2)$ -CMA-ES for 46 target values $\Delta f \in [10^{-8}, 10]$ in each dimension for functions f_1 – f_{24} . Markers on the upper or right edge indicate that the target value was never reached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

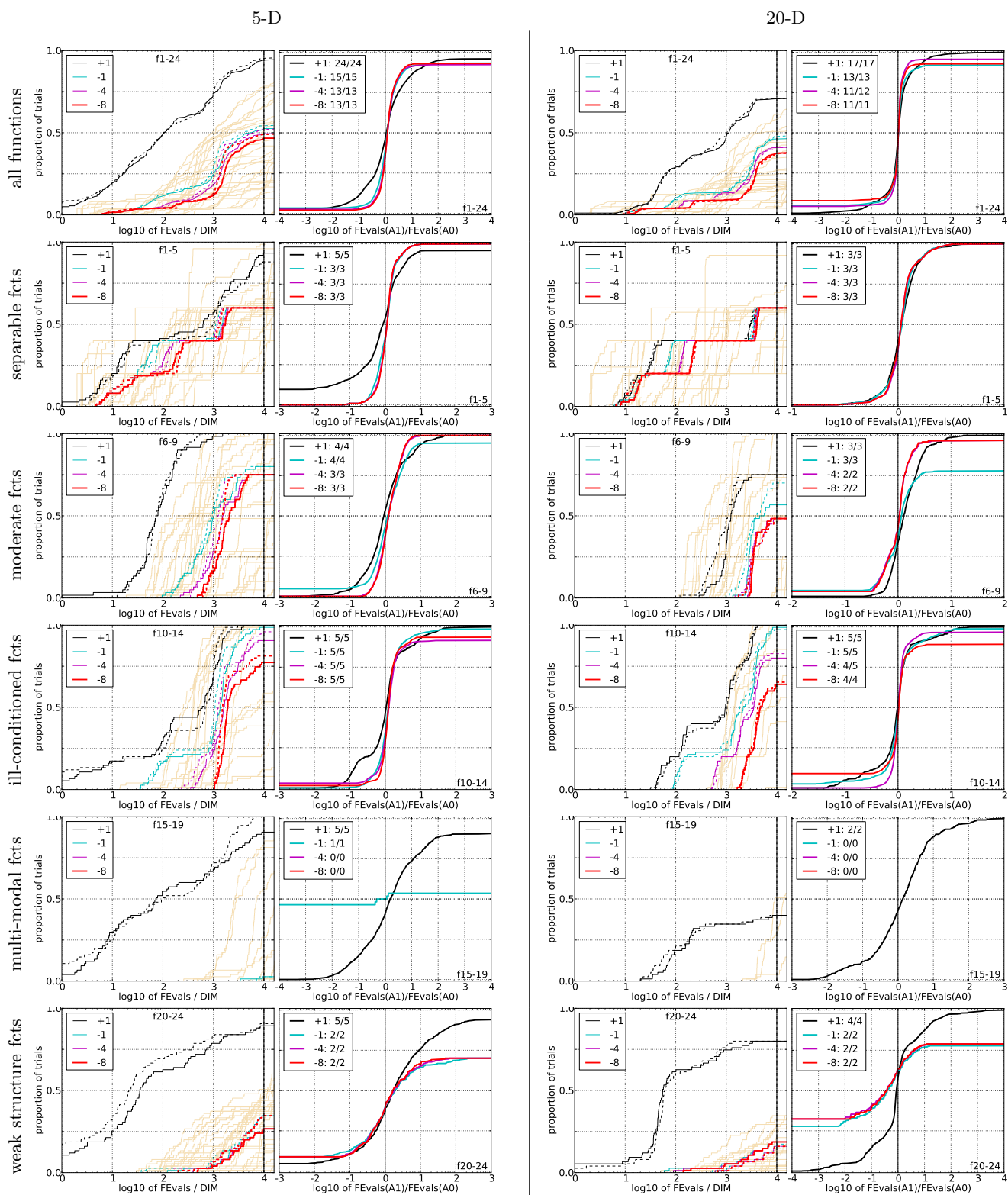


Figure 3: Empirical cumulative distributions (ECDF) of run lengths and speed-up ratios in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of function evaluations divided by dimension D (FEvals/ D) to reach a target value $f_{\text{opt}} + \Delta f$ with $\Delta f = 10^k$, where $k \in \{1, -1, -4, -8\}$ is given by the first value in the legend, for $(1, 2^s)$ -CMA-ES (solid) and $(1, 2)$ -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 $(1, 2^s)$ -CMA-ES divided by $(1, 2)$ -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 ($(1, 2^s)$ -CMA-ES first).

