

Benchmarking the (1+1)-CMA-ES on the BBOB-2009 Function Testbed

Anne Auger, Nikolaus Hansen

► **To cite this version:**

Anne Auger, Nikolaus Hansen. Benchmarking the (1+1)-CMA-ES on the BBOB-2009 Function Testbed. ACM-GECCO Genetic and Evolutionary Computation Conference, Jul 2009, Montreal, Canada. 2009. <inria-00430517>

HAL Id: inria-00430517

<https://hal.inria.fr/inria-00430517>

Submitted on 8 Nov 2009

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.

Benchmarking the (1+1)-CMA-ES on the BBOB-2009 Function Testbed

Anne Auger
TAO Team, INRIA Saclay Ile-de-France
LRI, Bat 490 Univ. Paris-Sud
91405 Orsay Cedex France
Anne.Auger@inria.fr

Nikolaus Hansen
Microsoft Research–INRIA Joint Centre
28 rue Jean Rostand
91893 Orsay Cedex, France
Nikolaus.Hansen@inria.fr

ABSTRACT

The (1+1)-CMA-ES is an adaptive stochastic algorithm for the optimization of objective functions defined on a continuous search space in a black-box scenario. In this paper, an independent restart version of the (1+1)-CMA-ES is implemented and benchmarked on the BBOB-2009 noise-free testbed. The maximum number of function evaluations per run is set to 10^4 times the search space dimension. The algorithm solves 23, 13 and 12 of 24 functions in dimension 2, 10 and 40, respectively.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Benchmarking, Black-box optimization, Evolutionary computation, CMA-ES

1. INTRODUCTION

Soon after the introduction of stochastic algorithms for optimizing functions [2], it was recognized that adaptive algorithms where the sampling distribution is adapted during the course of the optimization are necessary for optimizing efficiently. One pioneer work in this direction has been carried out by Schumer and Steiglitz in 1968 [11]. The algorithm they proposed is known in the field of evolutionary algorithms as the (1+1)-Evolution Strategy (ES) with one-fifth success rule and was independently discovered by others [10, 3]. In the (1+1)-ES with one-fifth success rule algorithm, the search distribution is spherical and one single

parameter of the distribution, the step-length (or step-size), is adapted based on the observed probability of success, defined as the probability to sample a solution better than the current solution.

Recently, the (1+1)-CMA-ES has been proposed as an extension of the (1+1)-ES with one-fifth success rule where not only a step-size but also a covariance matrix associated to the search distribution is adapted [8]. The covariance matrix adaptation (CMA) mechanism of the (1+1)-CMA-ES follows the principles introduced in [7].

In this paper, we introduce the (1+1)-CMA-ES with independent restarts and benchmark it on the BBOB-2009 noise-free function testbed.

2. THE (1+1)-CMA-ES WITH INDEPENDENT RESTARTS

In this section we describe the (1+1)-CMA-ES with independent restarts.

2.1 The (1+1)-CMA-ES

We start by describing the original (1+1)-CMA-ES as proposed in [8]. We consider an objective function $f : \mathbb{R}^D \mapsto \mathbb{R}$, $\mathbf{x} \mapsto f(\mathbf{x})$ to be minimized. The algorithm is given in Table 1 with time index n and in Table 2 without time index. In the following we describe the iteration n of the algorithm following Table 1. A candidate solution $\tilde{\mathbf{x}}_n$ is sampled by perturbing the current solution \mathbf{x}_n by adding a gaussian vector with mean vector $\mathbf{0}$ and covariance matrix \mathbf{C}_n and scaled by the step-size σ_n (Line 5)¹. This candidate solution is accepted if and only if $f(\tilde{\mathbf{x}}_n) \leq f(\mathbf{x}_n)$ (Line 9). The step-size is adapted (Lines 6 and 7) using the averaged success rate p_{n+1}^{succ} : it is increased if the success rate is strictly larger than the target probability $p_{\text{target}}^{\text{succ}}$, and decreased if it is strictly smaller, according to the equation in Line 7². If $f(\tilde{\mathbf{x}}_n) \leq f(\mathbf{x}_n)$, the covariance matrix is adapted by adding to a multiple of \mathbf{C}_n the rank-one update matrix $\mathbf{p}_{n+1}\mathbf{p}_{n+1}^T$ (Line 11) where the evolution path \mathbf{p}_{n+1} was updated in Line 10. If $f(\tilde{\mathbf{x}}_n) > f(\mathbf{x}_n)$, the covariance matrix is not changed, *i.e.*, \mathbf{C}_{n+1} equals \mathbf{C}_n . The evolution path is also not updated (Line 14). The default parameters for the (1+1)-CMA-ES are given in Table 3.

¹If \mathbf{A}_n satisfies $\mathbf{A}_n\mathbf{A}_n^T = \mathbf{C}_n$ and \mathbf{z} follows a multivariate normal distribution with mean vector $\mathbf{0}$ and covariance matrix \mathbf{I} , then $\mathbf{A}_n\mathbf{z}$ is a multivariate normal distribution with mean vector $\mathbf{0}$ and covariance matrix \mathbf{C}_n .

²For $c_p = 1$, $p_{\text{target}}^{\text{succ}} = 1/5$ one obtains the 1/5 success rule from [9] and benchmarked in [1] where the increase factor corresponds to $\exp(1/d)$.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.
Copyright 2009 ACM 978-1-60558-505-5/09/07 ...\$5.00.

Table 3: Default parameters for the (1+1)-CMA-ES

Step-size control:

$$d = 1 + \frac{D}{2}, p_{\text{target}}^{\text{succ}} = \frac{2}{11}, c_p = \frac{1}{12}$$

Covariance matrix adaptation:

$$c_c = \frac{2}{D+2}, c_{\text{cov}} = \frac{2}{D^2+6}, p^{\text{thresh}} = 0.44$$

2.2 The independent-restart (1+1)-CMA-ES

We have implemented an independent-restart version of the (1+1)-CMA-ES: for each start the initial solution \mathbf{x}_0 is sampled uniformly in $[-4, 4]^D$, the step-size σ_0 is initialized at 2 and the initial covariance matrix is the identity. After reaching a stopping criteria (described in the next section) the algorithm is (re-)initialized and restarted. This process is iterated. Whenever the overall number of function evaluations reaches $10^4 D$ or an objective function value below the target function value is reached the algorithm is stopped.

2.3 Termination criteria

A single run of the (1+1)-CMA-ES is terminated when one of the following condition is satisfied:

MaxIter = $100 + \lceil 1000D\sqrt{D} \rceil$ is the maximal number of iterations and function evaluations in each run of the (1+1)-CMA-ES

TolHistFun = 10^{-12} : the range of the best function values during the last $10 + 30D$ iterations is smaller than **TolHistFun**

TolX = 10^{-12} : all components of \mathbf{p}_n and all square roots of diagonal components of \mathbf{C}_n , multiplied by σ_n/σ_0 , are smaller than **TolX**

TolUpX = 10^8 : σ_n/σ_0 multiplied by the square root of the largest diagonal element of \mathbf{C}_n is larger than **TolUpX**

ConditionCov: the condition number of \mathbf{C}_n exceeds 10^{14} .

NoEffectAxis: \mathbf{x}_n remains numerically constant when adding $0.1\sigma_n\sqrt{l^t}\mathbf{v}^t$, where l^t is the $1 + (t \bmod D)$ -largest eigenvalue of \mathbf{C}_n and \mathbf{v}^t is the corresponding normalized eigenvector.

NoEffectCoor: any element of \mathbf{x}_n remains numerically constant when adding $0.2\sigma_n l^t$, where elements of l^t are the square root of the diagonal elements of \mathbf{C}_n .

3. PARAMETER TUNING

No parameter tuning has been conducted. The maximum number of iterations **MaxIter** has been set to prevent the possibility of excessively long runs. The other termination criteria were left to their standard values. The same parameter setting is used on all functions and therefore the crafting effort [5] computes to $\text{CrE} = 0$.

4. RESULTS

Results from experiments according to [5] on the benchmark functions given in [4, 6] are presented in Figures 1 and 2 and in Table 4.

In 5-D all the unimodal functions are solved and in 20-D and 40-D, the f_6 (Attractive sector) and f_7 (Step-ellipsoid)

functions are not solved. However, by decreasing the learning rate c_{cov} by a factor of 3, we have observed that the f_6 function could be solved. The decreased learning rate would increase the running time on ill-conditioned problems roughly by a factor of two. Among the multimodal functions, the (1+1)-CMA-ES can solve the f_{21} and f_{22} functions quite efficiently.

Compared to the (1+1)-ES with one-fifth success rule benchmarked in [1], the performance is greatly improved on all ill-conditioned problems that cannot be solved without the covariance matrix adaptation mechanism. The overall number of multimodal functions solved is the same for both algorithms, while CMA-ES is often faster. The (1+1)-CMA-ES is outperformed by the one-fifth success rule without CMA mechanism on the Attractive Sector function.

5. CPU TIMING EXPERIMENTS

For the timing experiment the (1+1)-CMA-ES restart was run with a maximum of $10^5 \times D$ function evaluations and restarted until 30 seconds has passed (according to Figure 2 in [5]). The experiments have been conducted with an Intel Pentium 4 CPU 3.80 GHz under Linux with Matlab R2007a. The time per function evaluation was 8.2; 8.3; 8.3; 8.7; 11; 20 times 10^{-4} seconds in dimensions 2; 3; 5; 10; 20; 40 respectively.

6. CONCLUSION

We have benchmarked the (1+1)-CMA-ES. The algorithm solves 23, 21, 16, 13, 12 and 12 of 24 functions in dimension 2, 3, 5, 10, 20 and 40, respectively, with a budget of $10^4 D$ function evaluations. The algorithm solves all unimodal functions in 5-D and all but 2 in 20 and 40-D. The good performances on the unimodal functions are to be expected from the adaptation mechanism of the algorithm inherited from the original CMA-ES [7]. Besides two exceptions the performance is poor on multi-modal functions due to the absence of a population in the algorithm. The results are expected to generalize well due to the invariance properties of the algorithm, namely invariance to order-preserving transformations of the function value and rotational invariance.

Acknowledgments

The authors would like to acknowledge Steffen Finck and Raymond Ros for their great and hard work on the BBOB project, Marc Schoenauer for his kind and persistent support and Dimo Brockhoff for his comments on the paper.

7. REFERENCES

- [1] A. Auger. Benchmarking the (1+1) Evolution Strategy with One-Fifth Success Rule on the BBOB-2009 Function Testbed. In *Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2009)*. ACM Press, 2009.
- [2] S. H. Brooks. A discussion of random methods for seeking maxima. *Operations Research*, 6:244–251, 1958.
- [3] L. Devroye. The compound random search. In *International Symposium on Systems Engineering and Analysis*, pages 195–110. Purdue University, 1972.

Table 1: Pseudo-code for the (1+1)-CMA-ES with time dependent notations

Algorithm: (1+1)-CMA-ES

1. Initialize $\mathbf{x}_0, \sigma_0, \mathbf{C}_0 = \mathbf{I}, \mathbf{p}_0 = \mathbf{0}, p_0^{\text{succ}} = p_{\text{target}}^{\text{succ}}, n = 1$
 2. **repeat**
 3. determine \mathbf{A}_n such that $\mathbf{C}_n = \mathbf{A}_n \mathbf{A}_n^T$
 4. $\mathbf{z}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ standard normal distribution
 5. $\tilde{\mathbf{x}}_n = \mathbf{x}_n + \sigma_n \mathbf{A}_n \mathbf{z}_n$ new solution
 6. $p_{n+1}^{\text{succ}} = (1 - c_p) p_n^{\text{succ}} + c_p \mathbf{1}_{\{f(\tilde{\mathbf{x}}_n) \leq f(\mathbf{x}_n)\}}$ averaged success rate
 7. $\sigma_{n+1} = \sigma_n \exp\left(\frac{1}{d} \frac{p_{n+1}^{\text{succ}} - p_{\text{target}}^{\text{succ}}}{1 - p_{\text{target}}^{\text{succ}}}\right)$ update step-size
 8. **if** $f(\tilde{\mathbf{x}}_n) \leq f(\mathbf{x}_n)$ **then**
 9. $\mathbf{x}_{n+1} = \tilde{\mathbf{x}}_n$
 10. $\mathbf{p}_{n+1} = (1 - c_c) \mathbf{p}_n + \mathbf{1}_{\{p_{n+1}^{\text{succ}} < p^{\text{thresh}}\}} \sqrt{c_c(2 - c_c)} \mathbf{A}_n \mathbf{z}_n$ update evolution path
 11. $\mathbf{C}_{n+1} = \left(1 - c_{\text{cov}} + c_{\text{cov}} \mathbf{1}_{\{p_{n+1}^{\text{succ}} > p^{\text{thresh}}\}} c_c(2 - c_c)\right) \mathbf{C}_n + c_{\text{cov}} \mathbf{p}_{n+1} \mathbf{p}_{n+1}^T$ update covariance matrix
 12. **else**
 13. $\mathbf{x}_{n+1} = \mathbf{x}_n$
 14. $\mathbf{p}_{n+1} = \mathbf{p}_n$
 15. $\mathbf{C}_{n+1} = \mathbf{C}_n$
 16. $n = n + 1$
 17. **until** stopping criterion is met
-

Table 2: Pseudo-code for the (1+1)-CMA-ES without time dependent notations

Algorithm: (1+1)-CMA-ES

1. Initialize $\mathbf{x}, \sigma, \mathbf{C} = \mathbf{I}, \mathbf{p} = \mathbf{0}, p^{\text{succ}} = p_{\text{target}}^{\text{succ}}$
 2. **repeat**
 3. determine \mathbf{A} such that $\mathbf{C} = \mathbf{A} \mathbf{A}^T$
 4. $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ standard normal distribution
 5. $\tilde{\mathbf{x}} = \mathbf{x} + \sigma \mathbf{A} \mathbf{z}$ new solution
 6. $p^{\text{succ}} \leftarrow (1 - c_p) p^{\text{succ}} + c_p \mathbf{1}_{\{f(\tilde{\mathbf{x}}) \leq f(\mathbf{x})\}}$ averaged success rate
 7. $\sigma \leftarrow \sigma \exp\left(\frac{1}{d} \frac{p^{\text{succ}} - p_{\text{target}}^{\text{succ}}}{1 - p_{\text{target}}^{\text{succ}}}\right)$ update step-size
 8. **if** $f(\tilde{\mathbf{x}}) \leq f(\mathbf{x})$ **then**
 9. $\mathbf{x} \leftarrow \tilde{\mathbf{x}}$
 10. $\mathbf{p} \leftarrow (1 - c_c) \mathbf{p} + \mathbf{1}_{\{p^{\text{succ}} < p^{\text{thresh}}\}} \sqrt{c_c(2 - c_c)} \mathbf{A} \mathbf{z}$ update evolution path
 11. $\mathbf{C} \leftarrow \left(1 - c_{\text{cov}} + c_{\text{cov}} \mathbf{1}_{\{p^{\text{succ}} > p^{\text{thresh}}\}} c_c(2 - c_c)\right) \mathbf{C} + c_{\text{cov}} \mathbf{p} \mathbf{p}^T$ update covariance matrix
 12. **until** stopping criterion is met
-

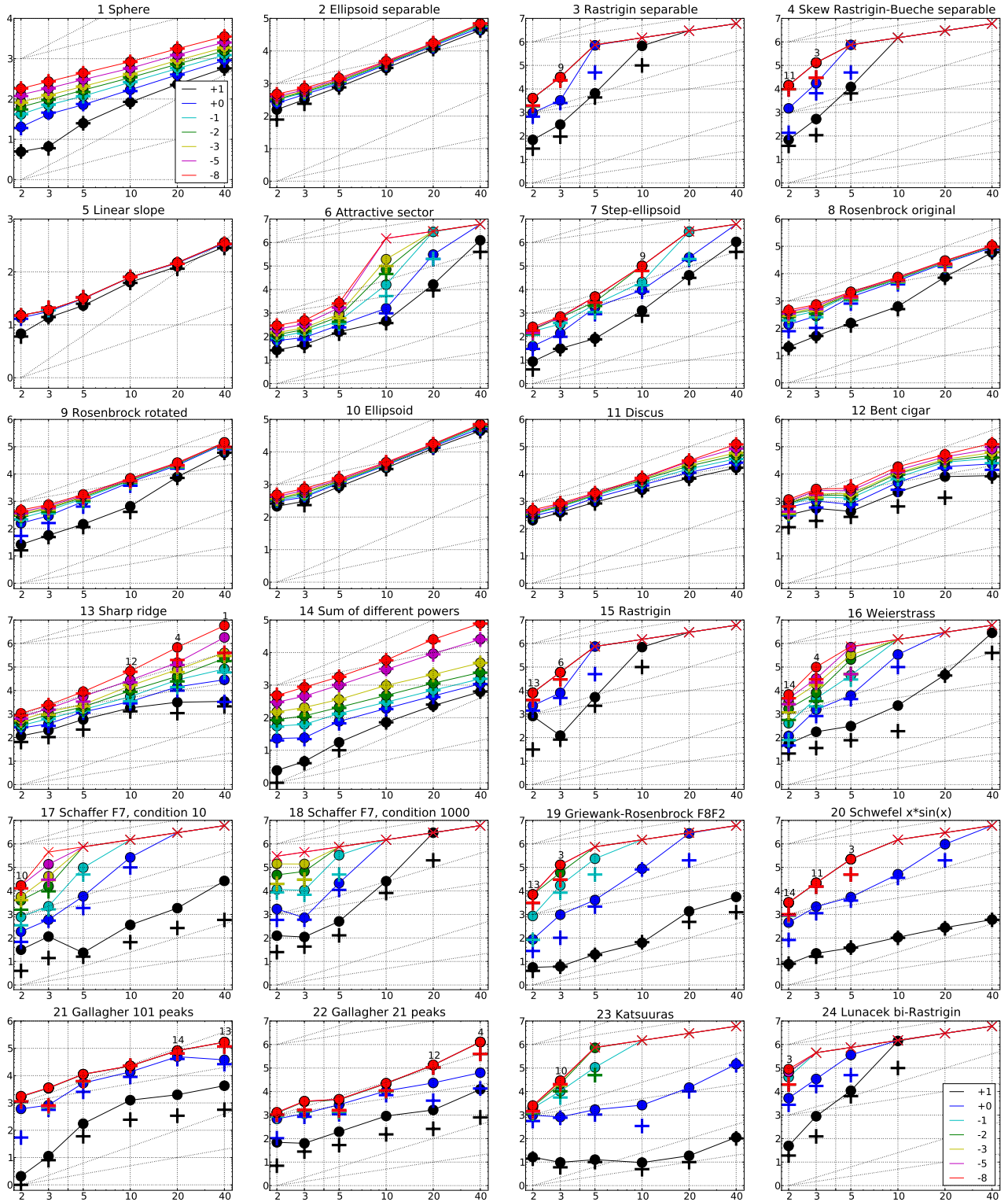


Figure 1: Expected Running Time (ERT, ●) to reach $f_{\text{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log presentation. The ERT(Δf) equals to $\#FEs(\Delta f)$ divided by the number of successful trials, where a trial is successful if $f_{\text{opt}} + \Delta f$ was surpassed during the trial. The $\#FEs(\Delta f)$ are the total number of function evaluations while $f_{\text{opt}} + \Delta f$ was not surpassed during the trial from all respective trials (successful and unsuccessful), and f_{opt} denotes the optimal function value. Crosses (×) indicate the total number of function evaluations $\#FEs(-\infty)$. Numbers above ERT-symbols indicate the number of successful trials. Annotated numbers on the ordinate are decimal logarithms. Additional grid lines show linear and quadratic scaling.

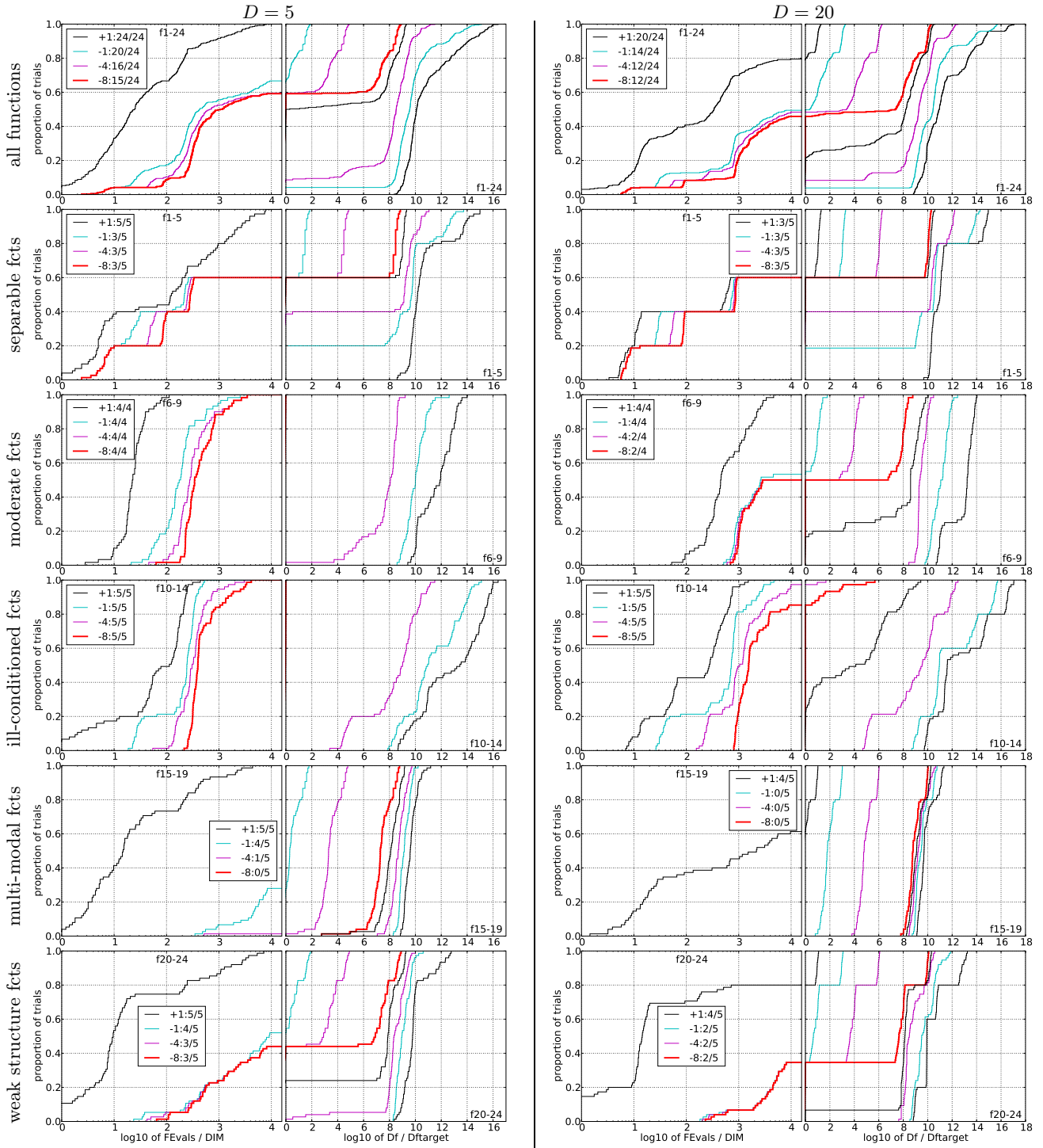


Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left subplots) or versus Δf (right subplots). The thick red line represents the best achieved results. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D , to fall below $f_{\text{opt}} + \Delta f$ 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 $D, 10D, 100D \dots$ function evaluations (from right to left cycling black-cyan-magenta). Top row: all results from all functions; second row: separable functions; third row: misc. moderate functions; fourth row: ill-conditioned functions; fifth row: multi-modal functions with adequate structure; last row: multi-modal functions with weak structure. 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.

- [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.
- [5] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA, 2009.
- [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.
- [7] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [8] C. Igel, T. Suttrop, and N. Hansen. A computational efficient covariance matrix update and a (1+1)-CMA for Evolution Strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006)*, pages 453–460. ACM Press, 2006.
- [9] S. Kern, S. Müller, N. Hansen, D. Büche, J. Ocenasek, and P. Koumoutsakos. Learning Probability Distributions in Continuous Evolutionary Algorithms - A Comparative Review. *Natural Computing*, 3(1):77–112, 2004.
- [10] I. Rechenberg. *Evolutionsstrategie*. Friedrich Frommann Verlag (Günther Holzboog KG), Stuttgart, 1973.
- [11] M. Schumer and K. Steiglitz. Adaptive step size random search. *Automatic Control, IEEE Transactions on*, 13:270–276, 1968.