

 Open access • Proceedings Article • DOI:10.1145/1570256.1570342

Benchmarking the (1+1) evolution strategy with one-fifth success rule on the BBOB-2009 function testbed — [Source link](#)

Anne Auger

Institutions: French Institute for Research in Computer Science and Automation

Published on: 08 Jul 2009 - Genetic and Evolutionary Computation Conference

Topics: Best-first search, Evolution strategy, Search algorithm, Benchmarking and Testbed

Related papers:

- [Adaptive step size random search](#)
- [Parameter Control in Evolutionary Algorithms: Trends and Challenges](#)
- [Learning Probability Distributions in Continuous Evolutionary Algorithms - a Comparative Review](#)
- [From black-box complexity to designing new genetic algorithms](#)
- [Parameter control in evolutionary algorithms](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/benchmarking-the-1-1-evolution-strategy-with-one-fifth-2xuss0ntc1>



HAL
open science

Benchmarking the (1+1) Evolution Strategy with One-Fifth Success Rule on the BBOB-2009 Function Testbed

Anne Auger

► **To cite this version:**

Anne Auger. Benchmarking the (1+1) Evolution Strategy with One-Fifth Success Rule on the BBOB-2009 Function Testbed. ACM-GECCO Genetic and Evolutionary Computation Conference, Jul 2009, Montreal, Canada. inria-00430515

HAL Id: inria-00430515

<https://hal.inria.fr/inria-00430515>

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) Evolution Strategy with One-Fifth Success Rule 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

ABSTRACT

In this paper, we benchmark the (1+1) Evolution Strategy (ES) with one-fifth success rule which is one of the first and simplest adaptive search algorithms proposed for optimization. The benchmarking is conducted on the noise-free BBOB-2009 testbed. We implement a restart version of the algorithm and conduct for each run 10^6 times the dimension of the search space function evaluations.

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, Evolution Strategies, One-Fifth success rule, adaptive search

1. INTRODUCTION

Soon after the introduction of the pure random search as a stochastic optimization algorithm [2], it was recognized that adaptive algorithms where the sampling distribution is adapted (as opposed to pure random search) during the course of the optimization are necessary for efficient optimization. One of the oldest adaptive search algorithms adapts the step-length (or step-size) using the following idea: the step-size is increased after a successful step and decreased after a failure so as to maintain a success probability of approximately $1/5$, increase the step-size if the success probability is larger than $1/5$ and decrease it otherwise. The

discovery of this idea, known in the field of evolutionary algorithms as the one-fifth success rule, dates back to 1968, introduced by Schumer and Steiglitz in [10] and discovered independently by others [9, 3].

In this paper, the (1+1) Evolution Strategy (ES) with one-fifth success rule and restart mechanism is benchmarked on the BBOB-2009 function testbed.

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

In this section we describe the (1+1)-ES with one-fifth success rule and independent restarts. We start by describing the (1+1)-ES with one-fifth success rule.

2.1 The (1+1)-ES with 1/5 success rule

The (1+1)-ES with one-fifth success rule implements the idea that the step-size should increase if “too many” steps are successful, indicating that the search is too local and should decrease if “too few” steps are successful, indicating that the step-length used for sampling solutions is “too large”. Optimally, the probability to sample successful steps should be close to one-fifth [10, 9].

The algorithm tested here and presented in Table 1 is the version of the one-fifth success rule presented in [8]. We consider a scalar objective function $f : \mathbb{R}^D \mapsto \mathbb{R}$, $\mathbf{x} \mapsto f(\mathbf{x})$ to be minimized and denote $\mathbf{X}_n \in \mathbb{R}^D$ the estimate of the solution (also called parent) at iteration n , and $\sigma_n \in \mathbb{R}_+$ the step-size. A new solution (or offspring) $\widetilde{\mathbf{X}}_n$ is sampled by adding to \mathbf{X}_n a spherical multivariate normal distribution centered at zero and scaled by the step-size σ_n (Table 1, line 3), *i.e.*,

$$\widetilde{\mathbf{X}}_n = \mathbf{X}_n + \sigma_n \mathcal{N}(\mathbf{0}, \mathbf{I}) ,$$

where $\mathcal{N}(\mathbf{0}, \mathbf{I})$ denotes a multivariate normal distribution with mean vector $\mathbf{0}$ and identity covariance matrix¹. The objective function value of $\widetilde{\mathbf{X}}_n$ is computed and \mathbf{X}_{n+1} equals $\widetilde{\mathbf{X}}_n$ if $f(\widetilde{\mathbf{X}}_n) \leq f(\mathbf{X}_n)$ or \mathbf{X}_n otherwise, *i.e.*, the best among offspring and parent is becoming the parent for the next iteration³.

¹The associated density is such that lines with equal density are hyperspheres.

²We assume minimization.

³This explains the notation (1+1) referring to the fact that at each iteration, there is a single parent (the first “1”), a single offspring (the second “1”) and the best, *i.e.*, the one having the smallest objective value, among the parent *plus* the offspring is kept for the next iteration.

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 1: Pseudo-code for the (1+1)-ES with one-fifth success rule.

Algorithm: (1+1)-ES with 1/5 success-rule

1. Initialize \mathbf{X}_0, σ_0
2. **repeat**
3. $\widetilde{\mathbf{X}}_n = \mathbf{X}_n + \sigma_n \mathcal{N}(\mathbf{0}, \mathbf{I})$ Sample one offspring
4. **if** $f(\widetilde{\mathbf{X}}_n) \leq f(\mathbf{X}_n)$ **then** If $f(\text{offsp.}) \leq f(\text{parent})$
5. $\mathbf{X}_{n+1} = \widetilde{\mathbf{X}}_n$ New parent = offsp.
6. $\sigma_{n+1} = 1.5 \sigma_n$ Step-size is increased
7. **else** If offspring strictly worse
8. $\mathbf{X}_{n+1} = \mathbf{X}_n$ New parent = old parent
9. $\sigma_{n+1} = 1.5^{-1/4} \sigma_n$ Step-size is decreased
10. **until** stopping criteria is met

The step-size is increased by a factor 1.5 if the offspring is successful, *i.e.*, if its objective function value is smaller or equal than the one of its parent (Table 1, line 6) and is decreased by a factor $1.5^{-1/4}$ otherwise (Table 1, line 9). We now sketch why such a choice for the increasing and decreasing factors implements the idea of the one-fifth success rule. If the probability to sample a successful offspring from a given parent with step-size σ_n is 1/5, then the probability to sample an unsuccessful offspring equals 4/5 and on average

$$E(\sigma_{n+1}|\sigma_n) = \left((1.5)^{-1/4} \right)^{4/5} (1.5)^{1/5} \sigma_n .$$

Since the term $\left((1.5)^{-1/4} \right)^{4/5} (1.5)^{1/5}$ equals 1 we hence have

$$E(\sigma_{n+1}|\sigma_n) = \sigma_n .$$

Furthermore, if the probability of sampling successful offspring, that we denote p_s is larger than 1/5, on average the step-size will satisfy

$$E(\sigma_{n+1}|\sigma_n) = \left((1.5)^{-1/4} \right)^{1-p_s} (1.5)^{p_s} \sigma_n ,$$

and thus the step-size will increase on average since the term

$$\left((1.5)^{-1/4} \right)^{1-p_s} (1.5)^{p_s}$$

is strictly larger 1. The same reasoning holds if p_s is smaller than 1/5 implying that the step-size will decrease on average.

We have tested two other variants that are given in Table 2 and Table 3. In Variant 1, in case of equality between the objective function values of the offspring and parent the step-size stays constant and the current estimate of the solution \mathbf{X}_n stays constant as well. In the second variant (Table 3), in case of equality between the objective function values of the offspring and parent, the step-size stays constant but the \mathbf{X}_{n+1} takes the value of the offspring $\widetilde{\mathbf{X}}_n$.

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

We have implemented an independent-restart version of the (1+1)-ES: for each start the initial solution \mathbf{X}_0 is sampled uniformly in $[-4, 4]^D$ and the step-size σ_0 is initialized at 2. After reaching a stopping criteria (described in the next section) the algorithm is (re-)initialized and restarted.

Table 2: Pseudo-code for the (1+1)-ES with one-fifth success rule Variant 1.

Algorithm: (1+1)-ES with 1/5 success-rule Variant 1

1. Initialize \mathbf{X}_0, σ_0
2. **repeat**
3. $\widetilde{\mathbf{X}}_n = \mathbf{X}_n + \sigma_n \mathcal{N}(\mathbf{0}, \mathbf{I})$
4. **if** $f(\widetilde{\mathbf{X}}_n) < f(\mathbf{X}_n)$ **then**
5. $\mathbf{X}_{n+1} = \widetilde{\mathbf{X}}_n$
6. $\sigma_{n+1} = 1.5 \sigma_n$
7. **elseif** $f(\widetilde{\mathbf{X}}_n) = f(\mathbf{X}_n)$ **then**
8. $\mathbf{X}_{n+1} = \mathbf{X}_n$
9. $\sigma_{n+1} = \sigma_n$
10. **else**
11. $\mathbf{X}_{n+1} = \mathbf{X}_n$
12. $\sigma_{n+1} = 1.5^{-1/4} \sigma_n$
13. **until** stopping criteria is met

Table 3: Pseudo-code for the (1+1)-ES with one-fifth success rule Variant 2.

Algorithm: (1+1)-ES with 1/5 success-rule Variant 2

1. Initialize \mathbf{X}_0, σ_0
2. **repeat**
3. $\widetilde{\mathbf{X}}_n = \mathbf{X}_n + \sigma_n \mathcal{N}(\mathbf{0}, \mathbf{I})$
4. **if** $f(\widetilde{\mathbf{X}}_n) < f(\mathbf{X}_n)$ **then**
5. $\mathbf{X}_{n+1} = \widetilde{\mathbf{X}}_n$
6. $\sigma_{n+1} = 1.5 \sigma_n$
7. **elseif** $f(\widetilde{\mathbf{X}}_n) = f(\mathbf{X}_n)$ **then**
8. $\mathbf{X}_{n+1} = \widetilde{\mathbf{X}}_n$
9. $\sigma_{n+1} = \sigma_n$
10. **else**
11. $\mathbf{X}_{n+1} = \mathbf{X}_n$
12. $\sigma_{n+1} = 1.5^{-1/4} \sigma_n$
13. **until** stopping criteria is met

This process is iterated. Whenever the overall number of function evaluations reaches $10^6 D$ or an objective function value below the target function value is reached, the algorithm is stopped.

2.3 Stopping criteria for single runs

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

- **TolSigma** = 10^{-15} : stop if $\sigma_n \leq \text{TolSigma}$
- **MaxNoImp** = $4 * 4 \frac{\ln(10)}{\ln(1.5)}$: stop if there was no improvements during **MaxNoImp** successive iterations (*i.e.* lower value found) of the objective function

3. PARAMETER TUNING

No specific parameter tuning has been done, several trials on the whole testbed have been done to determine the value for `To1Sigma` since some previous values set by the author turned out to be too large (typically 10^{-8} is too large) and did not allow to observe convergence on the Attractive Sector function. The same settings have been used for all functions such that the crafting effort [5] computes to $CrE = 0$.

4. CPU TIMING EXPERIMENT

For the timing experiment the (1+1)-ES with independent restarts was run with a maximum of 10^5 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 using a C-implementation. The time per function evaluation was 7.5; 9.9; 15; 27; 51; 97 times 10^{-7} seconds in dimensions 2; 3; 5; 10; 20; 40 respectively.

5. 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.

Among the 24 functions, 13 were solved in 5-D and 9 in 20-D. Not too surprisingly, the (1+1)-ES can solve all moderate functions in 5-D and 3 out of 4 in 20-D with an ERT of less than 10^6 D. For the ill-conditioned problems, the picture is different since only 2 out of 5 problems are solved in 20-D and 1 in 5-D. Furthermore solving ill-conditioned problems requires an ERT larger than 6×10^5 D though most of the moderate problems can be solved with an ERT smaller than 10^5 D. Besides, though theoretically, the (1+1)-ES should also solve the Ellipsoid function if the maximum number of evaluations would be large enough, we have observed that due to numerical precisions, the (1+1)-ES cannot solve it even after increasing the maximum number of evaluations. The bad results on ill-conditioned problems are of course not surprising because the method has no mechanism to deform the sampling distribution, as opposed to the famous Covariance Matrix Algorithm [7].

Furthermore, we can observe a consequence of the invariance of the algorithm with respect to the coordinate system: the performances are invariant on the original / rotated Ellipsoid, Rosenbrock, Rastrigin functions.

The performance is poor on multi-modal functions, however the Gallagher multimodal functions are solved.

The Variant 1 and 2 gave slightly worse results. In particular on the step-ellipsoid where increasing the step-size is beneficial for facing the plateaus.

6. DISCUSSION

The simple (1+1)-ES with one-fifth success rule with a restart mechanism is able to solve 13 (resp. 9) functions in 5-D (resp. 20-D) but performs poorly on ill-conditioned problems. This is to be expected from the lack of adaptation mechanism for the covariance matrix and can be improved by introducing such a mechanism [1]. The performance is poor on multi-modal functions. The results are foreseen to generalize well due to two invariance properties of the algorithm: invariance to order-preserving transformations of the function value and rotational invariance.

Acknowledgments

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

7. REFERENCES

- [1] A. Auger and N. Hansen. Benchmarking the (1+1)-CMA-ES 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.
- [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] 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.
- [9] I. Rechenberg. *Evolutionstrategie*. Friedrich Frommann Verlag (Günther Holzboog KG), Stuttgart, 1973.
- [10] M. Schumer and K. Steiglitz. Adaptive step size random search. *Automatic Control, IEEE Transactions on*, 13:270–276, 1968.

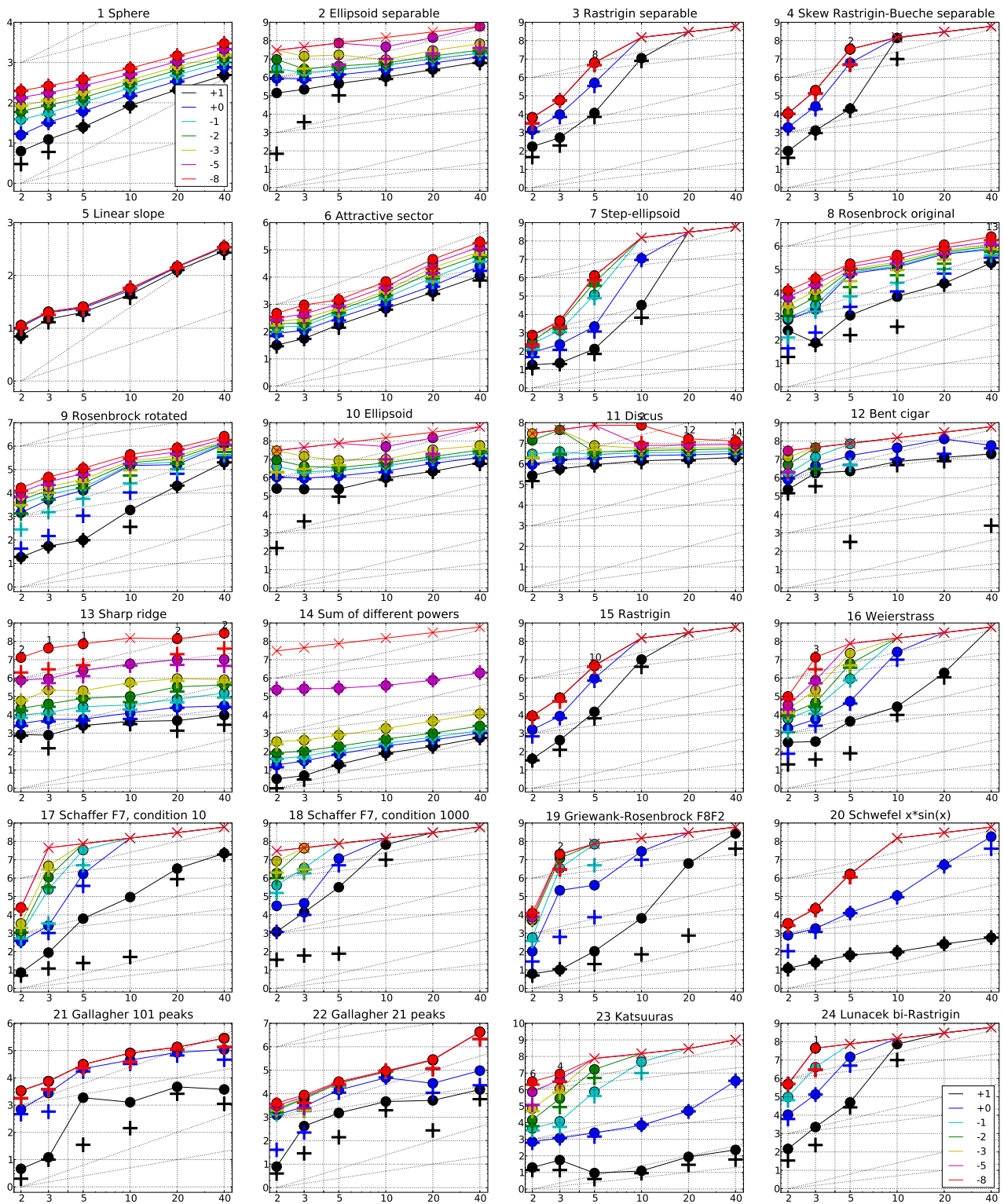


Figure 1: Expected Running Time (ERT, ●) to reach $f_{\text{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log 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.

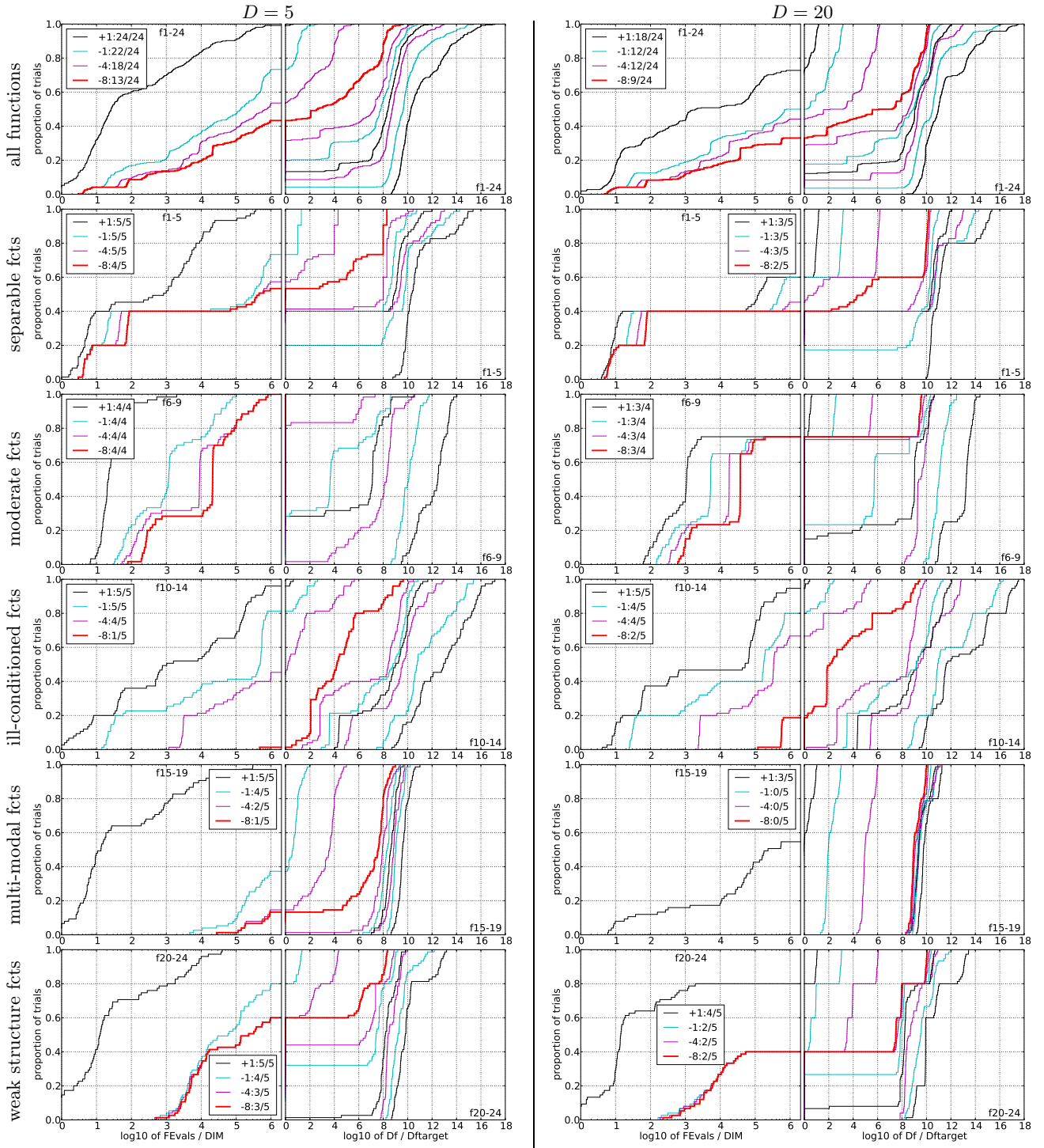


Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left subplots) or versus Δf (right subplots). The thick red line represents the best achieved results. Left subplots: ECDF of the running time (number of function evaluations), divided 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.