BBOB-Benchmarking a Simple Estimation of Distribution Algorithm with Cauchy Distribution

Petr Pošík
Czech Technical University in Prague, Faculty of Electrical Engineering, Dept. of Cybernetics
Technická 2, 166 27 Prague 6
posik@labe.felk.cvut.cz

ABSTRACT

The restarted estimation of distribution algorithm (EDA) with Cauchy distribution as the probabilistic model is tested on the BBOB 2009 testbed. These tests prove that when using the Cauchy distribution and suitably chosen variance enlargment factor, the algorithm is usable for broad range of fitness landscapes, which is not the case for EDA with Gaussian distribution which converges prematurely. The results of the algorithm are of mixed quality and its scaling is at least quadratic.

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, Experimentation, Performance, Reliability

Keywords

Benchmarking, Black-box optimization, Local search, Estimation-of-distribution algorithms, Evolutionary computation, Cauchy distribution

1. INTRODUCTION

Estimation of distribution algorithms (EDAs) [11] are a class of evolutionary algorithms (EAs) that do not use the crossover and mutation operators to create the offspring population. Instead, they build a probabilistic model describing the distribution of promising individuals and create offspring by sampling from the model. In real-valued spaces, such an algorithm can have a simple structure which is depicted in Fig. 1.

If the Gaussian distribution is employed as the model of promising individuals ([10], [3], [16], [1]), and the parameters of the distribution, μ, σ, and R, are learned by maximum likelihood (ML) estimation, the algorithm is very prone to premature convergence (i.e. the population converges on the slope of the fitness function) as recognized by many authors (see e.g. [3], [17], [12]). In [7], it was shown also theoretically that the distance traversed by a simple Gaussian EDA with truncation selection is bounded, and [5] showed similar results for tournament selection.

1. Initialize the parameters $\mu^0 = (\mu_1^0, \ldots, \mu_D^0)$, $\sigma^0 = (\sigma_1^0, \ldots, \sigma_D^0)$, and $R^0 \in \mathcal{R}^{D\times D}$, R is orthonormal, D is the dimensionality of the search space. Generation counter $t = 0$.
2. Sample N offspring from the search distribution (use R^t as a rotation matrix containing the base vectors, σ^t as relative scaling factors of individual components, and μ^t as the distribution center).
3. Evaluate the individuals.
4. Select the τN best solutions (truncation selection).
5. Estimate new μ^{t+1}, σ^{t+1}, and R^{t+1} using the selected individuals.
6. Enlarge the σ^{t+1} by a constant factor k (global step size).
7. Advance generation counter: $t = t + 1$.
8. If termination condition is not met, go to step 2.

Figure 1: Simple EDA used in this article.

Many techniques that fight the premature convergence were developed, usually by means of artificially enlarging the ML estimate of variance of the learned distribution. In [17] it is suggested to use standard deviation greater than 1 when sampling the base Gaussian distribution (before it is multiplied by $R\times\mathrm{diag}(\sigma)$), e.g. to use $\mathcal{G}(0, 1.5)$, which amounts to use the variance enlargment factor $k = 1.5$. Adaptive variance scaling (AVS), i.e. enlarging the variance when better solutions were found and shrinking the variance in case of no improvement, was used along with various techniques to trigger the AVS only on the slope of the fitness function in [6] and [2]. The algorithm in Fig. 1, that suggests enlarging the population variance by a constant factor each generation, was studied in [18] where the minimal values of the 'amplification coefficient' were determined by a search in 1D case.

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

In [13], the theoretical model of the algorithm behavior in 1D was used to derive the minimal and maximal admissible values for k. However, in [15] it was shown experimentaly that a constant multiplier does not ensure the desired properties of the algorithm when increasing the dimensionality of the search space.

In this article, we assess another premature convergence fighting technique suggested in [14], namely using the Cauchy distribution instead of Gaussian. This modification allows us to use a constant multiplier k which works for both the slope-like local neighborhood and the valley-like neighborhood in the same time.

2. ALGORITHM DESCRIPTION

The model parameters are computed actually using ML estimates for Gaussian distribution (even though they are subsequently used for the Cauchy distribution—they cannot be considered maximum-likelihood anymore, but they can serve as a heuristic). The distribution center μ is computed as the average of the selected data points, the rotation matrix R and the standard deviations σ are obtained by eigendecomposition of the covariance matrix of the selected data points.

The sampling process can be described as follows. First, for each i-th offspring a Cauchy distributed number r_i is sampled; it will be the radius, i.e. the distance of the offspring from the origin (for a negative radius, absolute value is used, even though it is not necessary). Then, the values of r_i are divided by the $\frac{1+\tau}{2}$-quantile of the Cauchy distribution ($\tau = 0.3$ is the used selection proportion of the truncation selection). It is just a form of standardizing the distribution used in [14]. Now, for each offspring a random vector is sampled, uniformly distributed on the unit sphere, which is then multiplied by the radius r_i. This operation results in an isotropically distributed set of offspring vectors z_i. Based on z_i, the final offspring x_i are generated by

$$x_i = \mu + k \cdot R \times \text{diag}(\sigma) \times z_i, \tag{1}$$

where k is the constant multiplier.

2.1 Parameter Settings

The algorithm has three parameters which determine its behaviour:

1. the selection proportion ($\tau = 0.3$ is used throughout this article),
2. the population size N (depends on the search space dimensionality), and
3. the variance enlarging factor k (also depends on the search space dimensionality).

By running a small experiment with the algorithm on the Rosenbrock's function it was observed, that the values of k and N shown in Fig. 1 give acceptable performace.

Based on these measurements, the following models for k and N were obtained:

$$N = 10^{1.05} \cdot D^{1.36} \tag{2}$$

$$k = \begin{cases} (0.3 + 0.25 \cdot D) \cdot \sqrt{D} & \text{if } D < 5, \\ (1.45 + 0.013 \cdot D) \cdot \sqrt{D} & \text{if } D \geq 5. \end{cases} \tag{3}$$

D	2	3	5	10	20
k	0.7	1.2	1.5	1.6	1.7
N	30	50	80	250	700

Table 1: Suggested values of parameters k and N for various values of D and for selection proportion $\tau = 0.3$.

In each iteration, a check is performed if the model standard deviations did not fall below an acceptable limit. The limit was set to 10^{-10}, and if smaller values of σ_i are detected, they are artificially enlarged to the least acceptable value.

2.2 Box Constraints? No

Several experiments with using box constraints in the form $\langle -5, 5\rangle^D$ or $\langle -6, 6\rangle^D$ were carried out. Subjectively, the best (the most regular) results were produced when the algorithm was run completely unconstrained, i.e. no constraints are used for the results presented in the next section.

2.3 The Crafting Effort

The algorithm has no additional parameters so that no further tuning is necessary; the same setup is used for all benchmark functions—the crafting effort CrE = 0.

2.4 Invariance Properties

Cauchy EDA is *invariant with respect to translation and rotation* as can be seen in the next section on the graphs for unrotated–rotated versions of Ellipsoid (2 and 10), Rastrigin (3 and 15), or Rosenbrock (8 and 9) functions. The algorithm uses fitness values only in the rank-based (truncation) selection, thus it is also *invariant against order-preserving transformations of the fitness function.*

3. EXPERIMENTAL PROCEDURE

The standard experimental procedure of BBOB was adopted: the algorithm was run on 24 test functions, 5 instances each, 3 runs on each instance. Each run was finished

- after finding a solution with fitness difference $\Delta f \leq 10^{-8}$, or
- after performing more than $5 \cdot 10^4 \times D$ function evaluations.

Each individual launch of the basic Cauchy EDA was interrupted (and the algorithm was restarted)

- after finding a solution with fitness difference $\Delta f \leq 10^{-8}$, or
- after performing more than the allowed number of function evaluations, or
- after the model converged too much, i.e. when $\max_i |\sigma_i| < 10^{-8}$.

4. RESULTS

Results from experiments according to [8] on the benchmark functions given in [4, 9] are presented in Figures 2 and 3 and in Table 2. The method solved at least 1 run (out of

15 runs) for 21 ($2D$), 16 ($3D$), 16 ($5D$), 15 ($10D$), and 10 ($20D$) out of 24 functions, respectively.

As it turned out, the effect of the restarting mechanism is not large since the vast majority of runs ended up with only 1 launch, i.e. the run either found the solution or was stopped by the maximum allowed number of evaluations. The third stopping criterion—stop if all σ are below certain threshold—was used only seldom.

5. CPU TIMING EXPERIMENT

The multistart algorithm was run with the maximal number of evaluations set to 10^5, the basic algorithm was restarted for at least 30 seconds. The average time demands were $5.1 \cdot 10^{-5}$, $1.7 \cdot 10^{-5}$, $9.4 \cdot 10^{-6}$, $8.6 \cdot 10^{-6}$, and $1.1 \cdot 10^{-5}$ seconds per function evaluation for 2-, 3-, 5-, 10-, and 20-dimensional search space, respectively. The experiment was conducted on Intel Core 2 CPU, T5600, 1.83 GHz, 1 GB RAM with Windows XP SP3 in MATLAB R2007b.

6. CONCLUSIONS

The restarted EDA with Cauchy distribution turned out to be effective for unimodal and ill-conditioned functions. On the other hand, its performance is bad when solving multimodal functions or functions with weak structure. Since it needs large number of function evaluations, it is likely that the restarting mechanism has only minor effect since only a few restarts were actually performed.

Acknowledgements

The author is supported by the Grant Agency of the Czech Republic with the grant no. 102/08/P094 entitled "Machine learning methods for solution construction in evolutionary algorithms".

7. REFERENCES

[1] C. W. Ahn, R. S. Ramakrishna, and D. E. Goldberg. Real-coded bayesian optimization algorithm: Bringing the strength of BOA into the continuous world. In *Proceedings of the Genetic and Evolutionary Computation Conference—GECCO 2004*, pages 840–851. Springer-Verlag, Berlin, 2004.

[2] P. A. N. Bosman, J. Grahl, and F. Rothlauf. SDR: A better trigger for adaptive variance scaling in normal EDAs. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and Evolutionary Computation*, pages 492–499, New York, NY, USA, 2007. ACM Press.

[3] P. A. N. Bosman and D. Thierens. Expanding from discrete to continuous estimation of distribution algorithms: The IDEA. In *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pages 767–776, London, UK, 2000. Springer-Verlag.

[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] C. Gonzales, J. A. Lozano, and P. Larrañaga. Mathematical modelling of UMDAc algorithm with tournament selection. *International Journal of Approximate Reasoning*, 31(3):313–340, 2002.

[6] J. Grahl, P. A. N. Bosman, and F. Rothlauf. The correlation-triggered adaptive variance scaling IDEA. In *Proceedings of the 8th annual conference on Genetic and Evolutionary Computation Conference – GECCO 2006*, pages 397–404, New York, NY, USA, 2006. ACM Press.

[7] J. Grahl, S. Minner, and F. Rothlauf. Behaviour of UMDAc with truncation selection on monotonous functions. In *IEEE Congress on Evolutionary Computation, CEC 2005*, volume 3, pages 2553–2559, 2005.

[8] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA, 2009.

[9] 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.

[10] P. Larrañaga, J. A. Lozano, and E. Bengoetxea. Estimation of distribution algorithms based on multivariate normal distributions and Gaussian networks. Technical Report KZZA-IK-1-01, Dept. of Computer Science and Artificial Intelligence, University of Basque Country, 2001.

[11] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms*. GENA. Kluwer Academic Publishers, 2002.

[12] J. Ocenasek, S. Kern, N. Hansen, and P. Koumoutsakos. A mixed bayesian optimization algorithm with variance adaptation. In X. Yao, editor, *Parallel Problem Solving from Nature - PPSN VIII*, pages 352–361. Springer-Verlag, Berlin, 2004.

[13] P. Pošík. Gaussian EDA and truncation selection: Setting limits for sustainable progress. In *IEEE SMC International Conference on Distributed Human-Machine Systems, DHMS 2008*, Athens, Greece, 2008. IEEE.

[14] P. Pošík. Preventing premature convergence in a simple EDA via global step size setting. In *Parallel Problem Solving from Nature Ű- PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 549–558. Springer, 2008.

[15] P. Pošík. Truncation selection and Gaussian EDA: Bounds for sustainable progress in high-dimensional spaces. In *EvoWorkshops 2008*, volume 4974 of *LNCS*, pages 525–534. Springer, 2008.

[16] S. Rudlof and M. Köppen. Stochastic hill climbing by vectors of normal distributions. In *First Online Workshop on Soft Computing*, Nagoya, Japan, 1996.

[17] B. Yuan and M. Gallagher. On the importance of diversity maintenance in estimation of distribution algorithms. In H. G. Beyer and U. M. O'Reilly, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2005*, volume 1, pages 719–726, New York, NY, USA, 2005. ACM Press.

[18] B. Yuan and M. Gallagher. A mathematical modelling technique for the analysis of the dynamics of a simple continuous EDA. In *IEEE Congress on Evolutionary Computation, CEC 2006*, pages 1585–1591, Vancouver, Canada, 2006. IEEE Press.

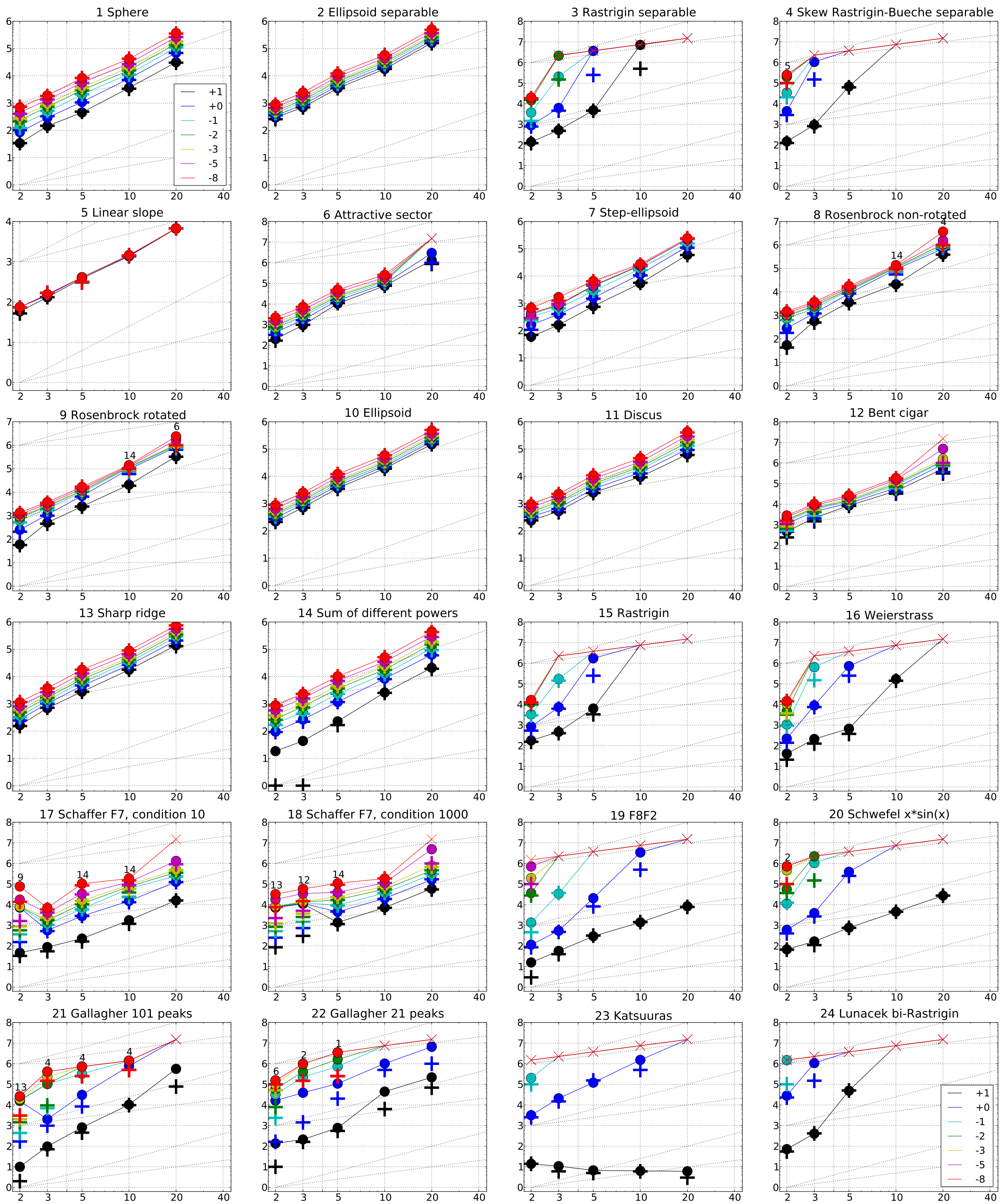

Figure 2: Expected Running Time (ERT, ●) to reach $f_{\rm opt} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log presentation. The $\mathrm{ERT}(\Delta f)$ equals to $\#\mathrm{FEs}(\Delta f)$ divided by the number of successful trials, where a trial is successful if $f_{\rm opt} + \Delta f$ was surpassed during the trial. The $\#\mathrm{FEs}(\Delta f)$ are the total number of function evaluations while $f_{\rm opt} + \Delta f$ was not surpassed during the trial from all respective trials (successful and unsuccessful), and $f_{\rm opt}$ denotes the optimal function value. Crosses ($\times$) indicate the total number of function evaluations $\#\mathrm{FEs}(-\infty)$. 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.

Δf	f_1 in 5-D, N=15, mFE=9660					f_1 in 20-D, N=15, mFE=454488				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	4.5e2	3.8e2	5.3e2	4.5e2	15	3.1e4	2.9e4	3.4e4	3.1e4
1	15	1.1e3	9.8e2	1.2e3	1.1e3	15	7.0e4	6.6e4	7.4e4	7.0e4
1e−1	15	2.1e3	1.9e3	2.2e3	2.1e3	15	1.1e5	1.0e5	1.1e5	1.1e5
1e−3	15	3.8e3	3.7e3	4.0e3	3.8e3	15	1.9e5	1.7e5	2.0e5	1.9e5
1e−5	15	5.7e3	5.5e3	5.8e3	5.7e3	15	2.6e5	2.5e5	2.8e5	2.6e5
1e−8	15	8.3e3	8.0e3	8.5e3	8.3e3	15	3.7e5	3.6e5	3.9e5	3.7e5

Δf	f_2 in 5-D, N=15, mFE=14280					f_2 in 20-D, N=15, mFE=602040				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	3.5e3	3.3e3	3.7e3	3.5e3	15	1.6e5	1.5e5	1.7e5	1.6e5
1	15	4.3e3	4.0e3	4.5e3	4.3e3	15	2.0e5	1.9e5	2.1e5	2.0e5
1e−1	15	5.1e3	4.8e3	5.4e3	5.1e3	15	2.4e5	2.2e5	2.5e5	2.4e5
1e−3	15	7.2e3	6.9e3	7.6e3	7.2e3	15	3.1e5	2.9e5	3.3e5	3.1e5
1e−5	15	9.3e3	9.0e3	9.7e3	9.3e3	15	3.9e5	3.7e5	4.1e5	3.9e5
1e−8	15	1.2e4	1.2e4	1.2e4	1.2e4	15	5.1e5	4.9e5	5.3e5	5.1e5

Δf	f_3 in 5-D, N=15, mFE=250005					f_3 in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	4.8e3	4.2e3	5.5e3	4.8e3	0	*69e+0*	*62e+0*	*74e+0*	5.6e5
1	1	3.6e6	3.5e6	3.8e6	1.4e5	.	.	.	.	.
1e−1	0	*26e-1*	*14e-1*	*35e-1*	1.0e5	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_4 in 5-D, N=15, mFE=250005					f_4 in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	6.9e4	5.6e4	8.2e4	6.9e4	0	*11e+1*	*89e+0*	*12e+1*	5.6e5
1	0	*78e-1*	*54e-1*	*91e-1*	7.1e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_5 in 5-D, N=15, mFE=1050					f_5 in 20-D, N=15, mFE=11832				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	3.9e2	3.0e2	4.7e2	3.9e2	15	6.6e3	5.9e3	7.4e3	6.6e3
1	15	4.1e2	3.3e2	5.1e2	4.1e2	15	6.7e3	5.9e3	7.5e3	6.7e3
1e−1	15	4.1e2	3.2e2	5.0e2	4.1e2	15	6.7e3	5.9e3	7.6e3	6.7e3
1e−3	15	4.1e2	3.3e2	5.0e2	4.1e2	15	6.7e3	5.9e3	7.5e3	6.7e3
1e−5	15	4.1e2	3.3e2	5.0e2	4.1e2	15	6.7e3	5.9e3	7.5e3	6.7e3
1e−8	15	4.1e2	3.2e2	5.1e2	4.1e2	15	6.7e3	6.0e3	7.5e3	6.7e3

Δf	f_6 in 5-D, N=15, mFE=59430					f_6 in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	1.0e4	9.1e3	1.2e4	1.0e4	10	1.3e6	1.3e6	1.4e6	8.8e5
1	15	1.5e4	1.3e4	1.6e4	1.5e4	5	3.0e6	3.0e6	3.0e6	9.9e5
1e−1	15	1.9e4	1.8e4	2.0e4	1.9e4	0	*17e-1*	*61e-2*	*21e+0*	8.9e5
1e−3	15	2.7e4	2.6e4	2.9e4	2.7e4	.	.	.	.	.
1e−5	15	3.7e4	3.5e4	3.9e4	3.7e4	.	.	.	.	.
1e−8	15	4.9e4	4.7e4	5.1e4	4.9e4	.	.	.	.	.

Δf	f_7 in 5-D, N=15, mFE=9450					f_7 in 20-D, N=15, mFE=386280				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	7.9e2	6.9e2	8.9e2	7.9e2	15	6.0e4	5.2e4	6.9e4	6.0e4
1	15	1.6e3	1.5e3	1.7e3	1.6e3	15	1.2e5	1.1e5	1.3e5	1.2e5
1e−1	15	2.8e3	2.6e3	3.0e3	2.8e3	15	1.7e5	1.6e5	1.8e5	1.7e5
1e−3	15	4.5e3	4.1e3	4.9e3	4.5e3	15	2.2e5	2.1e5	2.4e5	2.2e5
1e−5	15	4.5e3	4.1e3	4.9e3	4.5e3	15	2.2e5	2.1e5	2.4e5	2.2e5
1e−8	15	6.5e3	6.2e3	6.9e3	6.5e3	15	2.4e5	2.2e5	2.6e5	2.4e5

Δf	f_8 in 5-D, N=15, mFE=22890					f_8 in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	3.6e3	3.1e3	4.1e3	3.6e3	15	3.9e5	3.5e5	4.2e5	3.9e5
1	15	8.5e3	8.0e3	9.1e3	8.5e3	15	6.9e5	6.5e5	7.3e5	6.9e5
1e−1	15	1.1e4	1.1e4	1.2e4	1.1e4	14	8.5e5	8.0e5	8.9e5	8.0e5
1e−3	15	1.3e4	1.3e4	1.4e4	1.3e4	12	1.1e6	1.1e6	1.2e6	8.8e5
1e−5	15	1.5e4	1.5e4	1.6e4	1.5e4	9	1.6e6	1.5e6	1.6e6	9.1e5
1e−8	15	1.8e4	1.7e4	1.9e4	1.8e4	4	3.7e6	3.7e6	3.7e6	9.9e5

Δf	f_9 in 5-D, N=15, mFE=20475					f_9 in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	2.5e3	2.1e3	2.8e3	2.5e3	15	3.3e5	3.1e5	3.5e5	3.3e5
1	15	6.9e3	6.2e3	7.6e3	6.9e3	12	8.4e5	7.6e5	9.3e5	6.9e5
1e−1	15	9.7e3	9.0e3	1.0e4	9.7e3	12	9.5e5	8.9e5	1.0e6	7.8e5
1e−3	15	1.2e4	1.2e4	1.3e4	1.2e4	12	1.1e6	1.0e6	1.1e6	8.8e5
1e−5	15	1.4e4	1.3e4	1.5e4	1.4e4	8	1.7e6	1.6e6	1.8e6	9.0e5
1e−8	15	1.7e4	1.6e4	1.8e4	1.7e4	6	2.4e6	2.3e6	2.4e6	9.4e5

Δf	f_{10} in 5-D, N=15, mFE=14280					f_{10} in 20-D, N=15, mFE=559584				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	3.7e3	3.4e3	3.9e3	3.7e3	15	1.5e5	1.4e5	1.6e5	1.5e5
1	15	4.5e3	4.3e3	4.7e3	4.5e3	15	1.9e5	1.8e5	2.0e5	1.9e5
1e−1	15	5.4e3	5.1e3	5.6e3	5.4e3	15	2.2e5	2.1e5	2.3e5	2.2e5
1e−3	15	7.3e3	7.0e3	7.6e3	7.3e3	15	3.0e5	2.8e5	3.1e5	3.0e5
1e−5	15	9.3e3	9.0e3	9.5e3	9.3e3	15	3.7e5	3.5e5	3.8e5	3.7e5
1e−8	15	1.2e4	1.2e4	1.3e4	1.2e4	15	4.8e5	4.6e5	4.9e5	4.8e5

Δf	f_{11} in 5-D, N=15, mFE=12495					f_{11} in 20-D, N=15, mFE=574896				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	2.5e3	2.3e3	2.8e3	2.5e3	15	6.4e4	6.0e4	6.9e4	6.4e4
1	15	3.4e3	3.2e3	3.7e3	3.4e3	15	9.8e4	9.1e4	1.1e5	9.8e4
1e−1	15	4.6e3	4.4e3	4.8e3	4.6e3	15	1.4e5	1.3e5	1.5e5	1.4e5
1e−3	15	6.3e3	6.0e3	6.6e3	6.3e3	15	2.2e5	2.1e5	2.3e5	2.2e5
1e−5	15	8.2e3	7.8e3	8.5e3	8.2e3	15	3.0e5	2.8e5	3.2e5	3.0e5
1e−8	15	1.1e4	1.1e4	1.1e4	1.1e4	15	4.3e5	4.0e5	4.6e5	4.3e5

Δf	f_{12} in 5-D, N=15, mFE=38745					f_{12} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	8.6e3	7.9e3	9.2e3	8.6e3	13	5.3e5	4.2e5	6.4e5	4.9e5
1	15	1.1e4	1.0e4	1.2e4	1.1e4	10	8.6e5	7.0e5	1.0e6	5.7e5
1e−1	15	1.3e4	1.2e4	1.4e4	1.3e4	9	1.1e6	9.9e5	1.3e6	6.7e5
1e−3	15	1.8e4	1.6e4	1.9e4	1.8e4	8	1.6e6	1.5e6	1.7e6	8.6e5
1e−5	15	2.2e4	2.0e4	2.4e4	2.2e4	3	4.9e6	4.8e6	5.0e6	1.0e6
1e−8	15	2.7e4	2.5e4	2.9e4	2.7e4	0	*52e-5*	*63e-8*	*11e+0*	8.9e5

Δf	f_{13} in 5-D, N=15, mFE=20265					f_{13} in 20-D, N=15, mFE=824064				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	2.8e3	2.6e3	2.9e3	2.8e3	15	1.4e5	1.3e5	1.5e5	1.4e5
1	15	4.6e3	4.4e3	4.8e3	4.6e3	15	2.1e5	2.0e5	2.2e5	2.1e5
1e−1	15	6.3e3	6.1e3	6.6e3	6.3e3	15	2.8e5	2.7e5	2.9e5	2.8e5
1e−3	15	9.7e3	9.4e3	1.0e4	9.7e3	15	4.3e5	4.1e5	4.4e5	4.3e5
1e−5	15	1.3e4	1.3e4	1.3e4	1.3e4	15	5.6e5	5.4e5	5.8e5	5.6e5
1e−8	15	1.8e4	1.7e4	1.8e4	1.8e4	15	7.4e5	7.2e5	7.7e5	7.4e5

Δf	f_{14} in 5-D, N=15, mFE=12180					f_{14} in 20-D, N=15, mFE=604824				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	2.3e2	1.7e2	2.9e2	2.3e2	15	2.1e4	1.9e4	2.3e4	2.1e4
1	15	1.2e3	1.0e3	1.3e3	1.2e3	15	6.3e4	5.8e4	6.8e4	6.3e4
1e−1	15	2.3e3	2.1e3	2.5e3	2.3e3	15	1.1e5	9.7e4	1.2e5	1.1e5
1e−3	15	4.5e3	4.3e3	4.8e3	4.5e3	15	2.0e5	1.8e5	2.1e5	2.0e5
1e−5	15	7.0e3	6.7e3	7.3e3	7.0e3	15	3.0e5	2.8e5	3.1e5	3.0e5
1e−8	15	1.0e4	9.8e3	1.0e4	1.0e4	15	4.5e5	4.2e5	4.7e5	4.5e5

Δf	f_{15} in 5-D, N=15, mFE=250005					f_{15} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	6.4e3	4.5e3	8.4e3	6.4e3	0	*67e+0*	*60e+0*	*73e+0*	5.6e5
1	2	1.7e6	1.6e6	1.9e6	2.5e5	.	.	.	.	.
1e−1	0	*24e-1*	*81e-2*	*33e-1*	1.3e5	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{16} in 5-D, N=15, mFE=250005					f_{16} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	6.7e2	4.2e2	9.5e2	6.7e2	0	*16e+0*	*14e+0*	*19e+0*	4.0e5
1	4	7.3e5	6.1e5	8.4e5	2.5e5	.	.	.	.	.
1e−1	0	*15e-1*	*43e-2*	*25e-1*	1.1e5	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{17} in 5-D, N=15, mFE=250005					f_{17} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	2.3e2	1.5e2	3.0e2	2.3e2	15	1.6e4	1.2e4	2.0e4	1.6e4
1	15	2.8e3	2.5e3	3.1e3	2.8e3	15	1.3e5	1.2e5	1.4e5	1.3e5
1e−1	15	6.3e3	5.7e3	6.9e3	6.3e3	15	2.5e5	2.2e5	2.8e5	2.5e5
1e−3	15	1.6e4	1.5e4	1.6e4	1.6e4	15	4.9e5	4.6e5	5.2e5	4.9e5
1e−5	15	3.4e4	3.1e4	3.7e4	3.4e4	10	1.3e6	1.3e6	1.4e6	8.9e5
1e−8	14	1.0e5	8.5e4	1.2e5	1.0e5	0	*37e-7*	*20e-7*	*55e-6*	8.9e5

Δf	f_{18} in 5-D, N=15, mFE=250005					f_{18} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	1.4e3	1.2e3	1.6e3	1.4e3	15	5.9e4	5.3e4	6.6e4	5.9e4
1	15	4.5e3	4.3e3	4.8e3	4.5e3	15	1.7e5	1.5e5	1.8e5	1.7e5
1e−1	15	9.6e3	9.0e3	1.0e4	9.6e3	15	2.9e5	2.8e5	3.2e5	2.9e5
1e−3	15	2.5e4	2.3e4	2.7e4	2.5e4	13	7.8e5	7.2e5	8.5e5	7.1e5
1e−5	15	4.0e4	3.7e4	4.3e4	4.0e4	3	4.9e6	4.8e6	5.0e6	1.0e6
1e−8	14	1.1e5	8.9e4	1.3e5	1.0e5	0	*25e-6*	*34e-7*	*70e-4*	8.9e5

Δf	f_{19} in 5-D, N=15, mFE=250005					f_{19} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	3.0e2	2.3e2	3.7e2	3.0e2	15	8.4e3	7.3e3	9.5e3	8.4e3
1	15	2.1e4	1.4e4	2.9e4	2.1e4	0	*34e-1*	*31e-1*	*37e-1*	4.5e5
1e−1	0	*48e-2*	*22e-2*	*63e-2*	1.3e5	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{20} in 5-D, N=15, mFE=250005					f_{20} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	7.7e2	6.4e2	8.8e2	7.7e2	15	2.8e4	2.4e4	3.1e4	2.8e4
1	7	3.9e5	3.3e5	4.6e5	1.7e5	0	*27e-1*	*26e-1*	*28e-1*	3.5e5
1e−1	0	*11e-1*	*82e-2*	*13e-1*	1.0e5	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{21} in 5-D, N=15, mFE=250005					f_{21} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	8.1e2	4.9e2	1.2e3	8.1e2	10	5.7e5	3.7e5	7.7e5	2.5e5
1	14	3.1e4	1.1e4	5.3e4	3.1e4	0	*32e-1*	*20e-1*	*24e+0*	4.0e5
1e−1	7	3.2e5	2.5e5	4.0e5	1.9e5	.	.	.	.	.
1e−3	4	7.2e5	5.9e5	8.4e5	2.5e5	.	.	.	.	.
1e−5	4	7.2e5	6.1e5	8.4e5	2.5e5	.	.	.	.	.
1e−8	4	7.3e5	6.0e5	8.4e5	2.5e5	.	.	.	.	.

Δf	f_{22} in 5-D, N=15, mFE=250005					f_{22} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	7.7e2	6.1e2	9.5e2	7.7e2	13	2.2e5	8.4e4	3.6e5	2.1e5
1	11	1.1e5	6.3e4	1.6e5	5.6e4	2	6.6e6	5.7e6	7.3e6	1.0e6
1e−1	4	7.3e5	6.0e5	8.5e5	2.5e5	0	*51e-1*	*69e-2*	*11e+0*	4.0e5
1e−3	1	3.5e6	3.3e6	3.8e6	2.5e5	.	.	.	.	.
1e−5	1	3.5e6	3.3e6	3.8e6	2.5e5	.	.	.	.	.
1e−8	1	3.5e6	3.3e6	3.8e6	2.5e5	.	.	.	.	.

Δf	f_{23} in 5-D, N=15, mFE=250005					f_{23} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	6.7e0	4.7e0	9.1e0	6.7e0	15	6.1e0	4.0e0	8.2e0	6.1e0
1	14	1.2e5	8.9e4	1.5e5	1.2e5	0	*19e-1*	*14e-1*	*21e-1*	5.0e5
1e−1	0	*68e-2*	*49e-2*	*96e-2*	1.6e5	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{24} in 5-D, N=15, mFE=250005					f_{24} in 20-D, N=15, mFE=1000152				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	4.9e4	3.9e4	5.9e4	4.9e4	0	*91e+0*	*74e+0*	*97e+0*	6.3e5
1	0	*81e-1*	*66e-1*	*85e-1*	1.3e5	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 2: Shown are, for a given target difference to the optimal function value Δf: the number of successful trials (#); the expected running time to surpass $f_{\mathrm{opt}}+\Delta f$ (ERT, see Figure 2); the 10%-tile and 90%-tile of the bootstrap distribution of ERT; the average number of function evaluations in successful trials or, if none was successful, as last entry the median number of function evaluations to reach the best function value ($\mathrm{RT}_{\mathrm{succ}}$). If $f_{\mathrm{opt}}+\Delta f$ was never reached, figures in *italics* denote the best achieved Δf-value of the median trial and the 10% and 90%-tile trial. Furthermore, N denotes the number of trials, and mFE denotes the maximum of number of function evaluations executed in one trial. See Figure 2 for the names of functions.

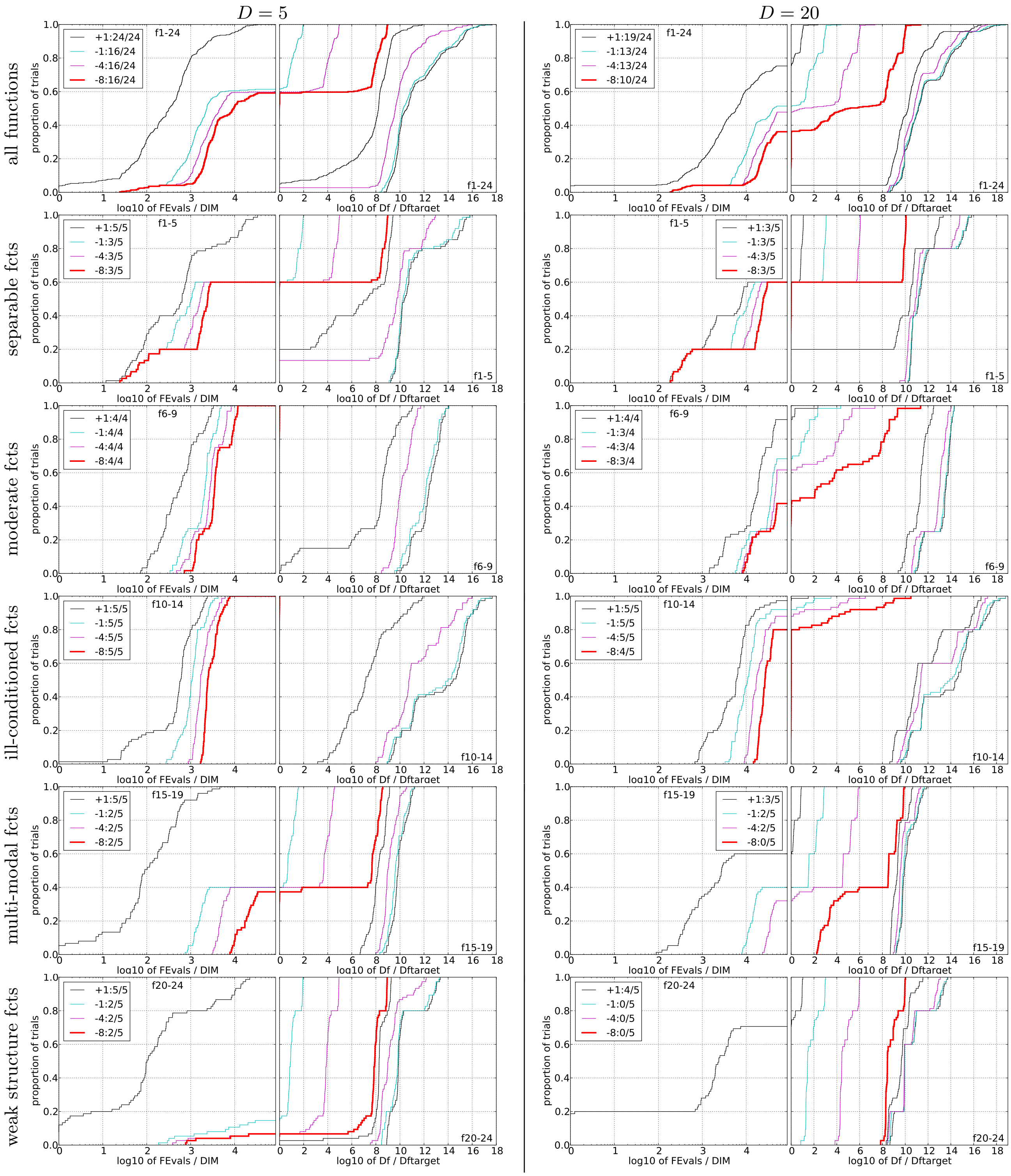

Figure 3: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left) or Δf. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\mathrm{opt}} + \Delta f$ with $\Delta f = 10^k$, 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 $D, 10\,D, 100\,D \ldots$ 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.