Benchmarking of MCS on the Noiseless Function Testbed

Waltraud Huyer
Fakultät für Mathematik
Universität Wien
Nordbergstraße 15
1090 Wien
Austria
Waltraud.Huyer@univie.ac.at

Arnold Neumaier
Fakultät für Mathematik
Universität Wien
Nordbergstraße 15
1090 Wien
Austria
Arnold.Neumaier@univie.ac.at

ABSTRACT

Benchmarking results with the MCS algorithm for bound-constrained global optimization on the noiseless BBOB 2009 testbed are described.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: OptimizationGlobal 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

Inspired by the DIRECT method by Jones et al. [5], the global optimization algorithm MCS (multilevel coordinate search) [4] was developed to minimize an objective function on a box $[u, v]$ with finite or infinite bounds. The algorithm proceeds by splitting the search space into smaller boxes, and each box contains a point whose function value is known. In the partitioning procedure parts where low function values are expected to be found are preferred. By starting a local search from certain good points, an improved result is obtained.

2. ALGORITHM PRESENTATION

Like DIRECT, the MCS algorithm combines global search (splitting boxes with large unexplored territory) and local search (splitting boxes with good function values). The key to balancing global and local search is the multilevel approach. As a rough measure of the number of times a box has been processed, a level $s \in \{1, \ldots, s_{\max}\}$ is assigned to each box, where boxes with level $s_{\max}$ are considered too small for further splitting. Whenever a box of level s $(0 < s < s_{\max})$ is split, its descendants get level $s + 1$ or $\min(s + 2, s_{\max})$. After an initialization procedure, the algorithm proceeds by a series of sweeps through the levels, i.e., it splits one box at each level, starting with the smallest non-empty level (i.e., with the largest boxes). We split along a single coordinate in each step, and information gained from already sampled points is used to determine the splitting coordinate as well as the position of the split.

MCS with local search tries to accelerate the convergence of the algorithm by starting local searches from the points belonging to boxes of level $s_{\max}$ before putting them into the so-called shopping basket (containing 'useful' points). The local search algorithm essentially consists of building a local quadratic model by triple searches, then defining a promising search direction by minimizing the quadratic model on a suitable box and finally making a line search along this direction.

The algorithm starts with a so-called initialization procedure producing an initial set of boxes. For each coordinate $i = 1, \ldots, n$, at least three values $x_i^1 < x_i^2 < \cdots < x_i^{L_i}$ in $[u_i, v_i]$, are needed, where n denotes the dimension of the problem and $L_i \geq 3$. Moreover, the pointers $l_i \in \{1, \ldots, L_i\}$ point to the initial point x^0, i.e., $x_i^0 = x_i^{l_i}$. The values x_i^j, $j = 1, \ldots, L_i$, l_i, and L_i, $i = 1, \ldots, n$, constitute the so-called initialization list.

The version of the software used can be downloaded from `http://www.mat.univie.ac.at/~neum/software/mcs/`.

3. EXPERIMENTAL PROCEDURE

For all control variables in the algorithm meaningful default values can be chosen that work simultaneously for most problems. MCS essentially contains the following parameters: the number $s_{\max}$ of levels, a limit $nf_{\max}$ on the overall number of function calls, an additional stopping criterion, the initialization list, and a limit nf_{local} on the number of function calls in a local search. We use the default value $s_{\max} = 5n + 10$, and the additional stopping criterion is given by reaching a target function value f_{target}.

Five kinds of initialization lists are incorporated into the MCS software. The safeguarded version for infinite box bounds was not considered since all the box bounds in our problems are finite, $u = (-5, \ldots, -5)^T$ and $v = (5, \ldots, 5)^T$. The default initialization list for finite u and v consists of boundary points and midpoint, with the midpoint as starting point, i.e., $L_i = 3$, $l_i = 2$, $x_i^1 = u_i$, $x_i^2 = \frac{1}{2}(u_i + v_i)$, and $x_i^3 = v_i$, $i = 1, \ldots, n$. Another initialization list for finite

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

bounds uses $x_i^1 = \frac{5}{6}u_i + \frac{1}{6}v_i$ and $x_i^3 = \frac{1}{6}u_i + \frac{5}{6}v_i$ instead of the boundaries (all other quantities are the same). There is also an option to generate an initialization list with the aid of line searches (described in detail Section 7.6 of [4]). We call the MCS algorithm with these three kinds of initialization lists MCS1, MCS2, and MCS3, respectively. Finally, it is possible to use a self-defined initialization list. After the initialization list has been chosen, MCS is purely deterministic, so the initialization list is the only possibility to introduce a random element in MCS.

In each call to MCS, we use $nf_{\max} = 500\max(n, 10)$ (i.e., $nf_{\max} = 5000$ for $n = 2, 3, 5, 10$ and $nf_{\max} = 10000$ for $n = 20$) and $nf_{\text{local}} = \text{round}(nf_{\max}/5)$, and $nf_{\max}$ might be slightly exceeded since the algorithm does not contain a check whether $nf_{\max}$ has been reached after each function call. Each trial consists of first applying the predefined initialization lists MCS1, MCS2, and MCS3 to the problem and then using a self-defined initialization list with $L_i = 3$, $l_i = 2$, and the values x_i^j, $j = 1, 2, 3$, drawn uniformly from $[u_i, v_i]$ for $i = 1, \ldots, n$ for at most 7 times for dimensions $n = 2, 3, 5$ and at most 5 times for dimensions $n = 10, 20$ (in order to save CPU time). I.e., each trial consists of at most 10 (or 8) attempts to solve the problem with MCS, and each call to MCS does not use any results from the previous calls. If the target function value f_{target} is reached, the trial is terminated and the subsequent calls to MCS are not made any more. So at most 50000 function calls (possibly a few more) are made in each trial for $n = 2, 3, 5$ and $4000\max(n, 10)$ for $n = 10, 20$. Three trials are made for the 5 function instances of each function. Note that if MCS1, MCS2, or MCS3 already solves the problem, the results are the same for the three trials since these choices of initialization lists do not contain a random element.

4. CPU TIMING EXPERIMENT

For the timing experiment according to [2], the experimental procedure described above was run on f_8 with at most 1000 function evaluations in each call to MCS and restarted until at least 30 seconds had passed. The timing experiment was carried out on an Intel Pentium 4 3.00 GHz under Ubuntu 4.0.3 with MATLAB 7.4.0.336, where most of the benchmarking tests were run. The results were 3.2, 2.0, 1.4, 1.1, 2.0, and 2.1 times 10^{-8} seconds per function evaluation in dimensions 2, 3, 5, 10, 20, and 40, respectively.

5. RESULTS

Results from experiments according to [2] on the benchmark functions given in [1, 3] are presented in Figures 1 and 2 and in Table 1.

6. REFERENCES

[1] 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.

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

[3] 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.

[4] W. Huyer and A. Neumaier. Global optimization by multilevel coordinate search. *J. Global Optimization*, 14:331–355, 1999.

[5] D. Jones, C. Perttunen, and B. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79:157–181, 1993.

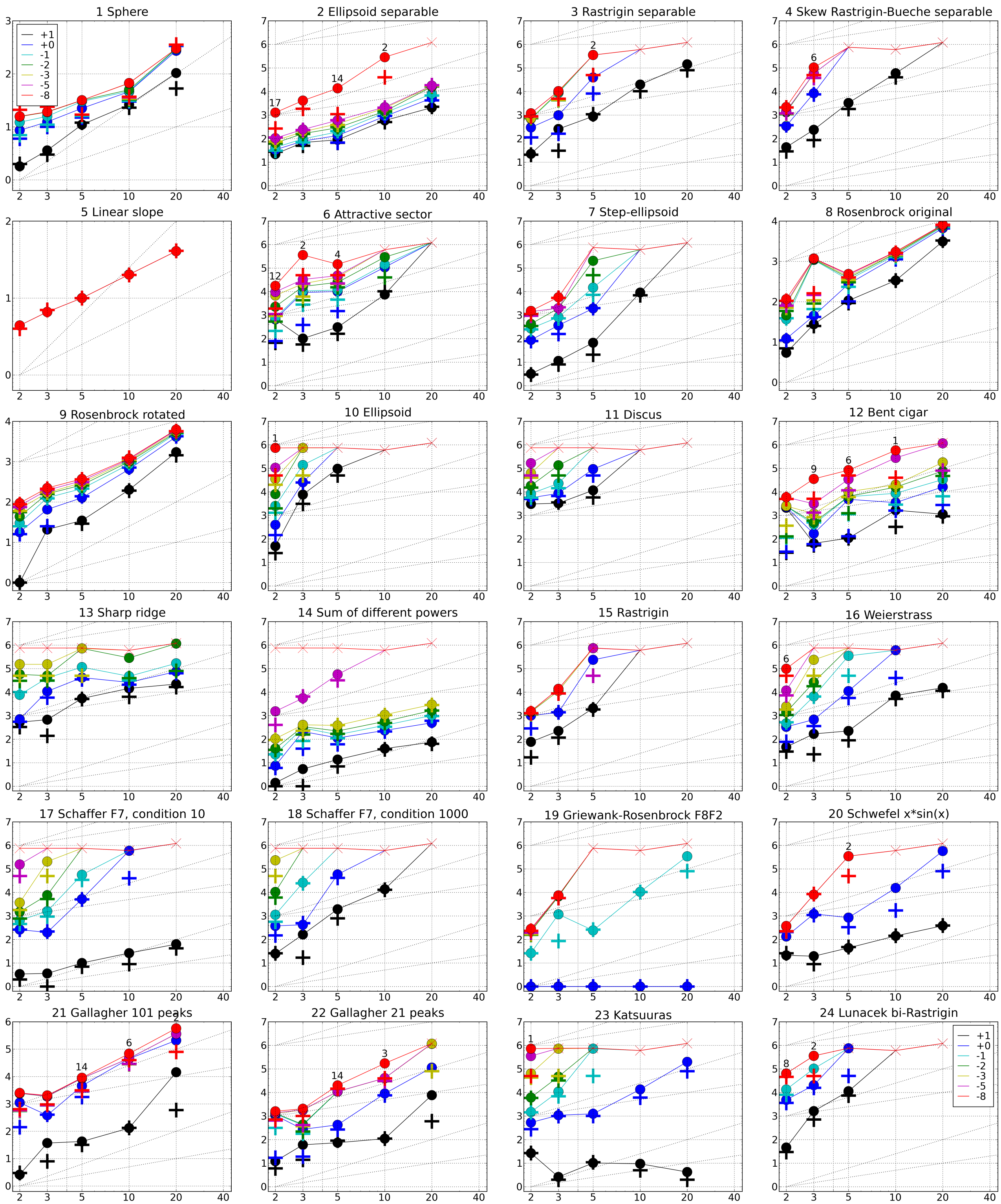

Figure 1: Expected Running Time (ERT, ●) to reach $f_{\mathrm{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log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

	f_1 in 5-D, N=15, mFE=60					**f_1 in 20-D**, N=15, mFE=396				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.1e1	1.1e1	1.1e1	1.1e1	15	1.0e2	7.2e1	1.4e2	1.0e2
1	15	2.2e1	1.7e1	2.8e1	2.2e1	15	2.7e2	2.3e2	3.1e2	2.7e2
1e−1	15	3.0e1	2.5e1	3.6e1	3.0e1	15	2.9e2	2.5e2	3.3e2	2.9e2
1e−3	15	3.2e1	2.7e1	3.7e1	3.2e1	15	3.0e2	2.6e2	3.4e2	3.0e2
1e−5	15	3.2e1	2.7e1	3.7e1	3.2e1	15	3.0e2	2.6e2	3.4e2	3.0e2
1e−8	15	3.2e1	2.7e1	3.7e1	3.2e1	15	3.0e2	2.6e2	3.4e2	3.0e2

	f_2 in 5-D, N=15, mFE=50006					**f_2 in 20-D**, N=15, mFE=81285				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	8.9e1	7.4e1	1.0e2	8.9e1	15	2.1e3	1.9e3	2.3e3	2.1e3
1	15	1.4e2	1.0e2	1.7e2	1.4e2	15	5.2e3	4.6e3	6.0e3	5.2e3
1e−1	15	1.9e2	1.7e2	2.2e2	1.9e2	15	8.0e3	6.7e3	9.4e3	8.0e3
1e−3	15	4.2e2	4.0e2	4.5e2	4.2e2	15	1.7e4	1.4e4	2.0e4	1.7e4
1e−5	15	6.0e2	5.6e2	6.4e2	6.0e2	15	1.8e4	1.5e4	2.1e4	1.8e4
1e−8	14	1.4e4	8.1e3	2.0e4	1.2e4	0	*30e-8*	*16e-8*	*41e-8*	2.8e4

	f_3 in 5-D, N=15, mFE=50682					**f_3 in 20-D**, N=15, mFE=80987				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	8.8e2	6.6e2	1.1e3	8.8e2	6	1.4e5	9.7e4	2.3e5	6.1e4
1	9	4.0e4	2.6e4	5.8e4	2.6e4	0	*13e+0*	*38e-1*	*23e+0*	2.5e4
1e−1	2	3.5e5	1.8e5	>7e5	5.0e4	.	.	.	.	.
1e−3	2	3.5e5	1.8e5	>7e5	5.0e4	.	.	.	.	.
1e−5	2	3.5e5	1.8e5	>7e5	5.0e4	.	.	.	.	.
1e−8	2	3.5e5	1.8e5	>7e5	5.0e4	.	.	.	.	.

	f_4 in 5-D, N=15, mFE=50462					**f_4 in 20-D**, N=15, mFE=80995				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	3.3e3	2.2e3	4.4e3	3.3e3	0	*21e+0*	*12e+0*	*50e+0*	2.8e4
1	0	*20e-1*	*20e-1*	*40e-1*	2.0e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_5 in 5-D, N=15, mFE=11					**f_5 in 20-D**, N=15, mFE=41				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.0e1	1.0e1	1.0e1	1.0e1	15	4.1e1	4.0e1	4.1e1	4.1e1
1	15	1.0e1	1.0e1	1.0e1	1.0e1	15	4.1e1	4.0e1	4.1e1	4.1e1
1e−1	15	1.0e1	1.0e1	1.0e1	1.0e1	15	4.1e1	4.0e1	4.1e1	4.1e1
1e−3	15	1.0e1	1.0e1	1.0e1	1.0e1	15	4.1e1	4.0e1	4.1e1	4.1e1
1e−5	15	1.0e1	1.0e1	1.0e1	1.0e1	15	4.1e1	4.0e1	4.1e1	4.1e1
1e−8	15	1.0e1	1.0e1	1.0e1	1.0e1	15	4.1e1	4.0e1	4.1e1	4.1e1

	f_6 in 5-D, N=15, mFE=50170					**f_6 in 20-D**, N=15, mFE=81002				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	3.0e2	1.9e2	4.2e2	3.0e2	0	*42e+0*	*23e+0*	*53e+0*	5.0e4
1	13	1.0e4	3.7e3	1.6e4	9.9e3	.	.	.	.	.
1e−1	13	1.2e4	5.5e3	1.8e4	1.1e4	.	.	.	.	.
1e−3	10	4.1e4	3.1e4	5.5e4	3.1e4	.	.	.	.	.
1e−5	9	4.8e4	3.4e4	6.9e4	2.9e4	.	.	.	.	.
1e−8	4	1.5e5	8.9e4	3.1e5	3.8e4	.	.	.	.	.

	f_7 in 5-D, N=15, mFE=50165					**f_7 in 20-D**, N=15, mFE=80253				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	6.7e1	4.0e1	9.5e1	6.7e1	0	*38e+0*	*23e+0*	*74e+0*	1.6e4
1	15	1.9e3	1.6e3	2.3e3	1.9e3	.	.	.	.	.
1e−1	13	1.5e4	9.0e3	2.1e4	1.5e4	.	.	.	.	.
1e−3	0	*25e-3*	*49e-4*	*24e-2*	4.0e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_8 in 5-D, N=15, mFE=918					**f_8 in 20-D**, N=15, mFE=14019				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.1e2	9.0e1	1.2e2	1.1e2	15	3.1e3	2.7e3	3.5e3	3.1e3
1	15	2.7e2	2.0e2	3.6e2	2.7e2	15	6.6e3	5.5e3	7.8e3	6.6e3
1e−1	15	3.5e2	2.7e2	4.3e2	3.5e2	15	7.2e3	6.1e3	8.4e3	7.2e3
1e−3	15	4.2e2	3.4e2	5.0e2	4.2e2	15	7.8e3	6.6e3	8.9e3	7.8e3
1e−5	15	4.6e2	3.8e2	5.4e2	4.6e2	15	7.9e3	6.8e3	9.1e3	7.9e3
1e−8	15	4.9e2	4.1e2	5.7e2	4.9e2	15	8.2e3	7.0e3	9.3e3	8.2e3

	f_9 in 5-D, N=15, mFE=424					**f_9 in 20-D**, N=15, mFE=8367				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	3.5e1	3.0e1	3.9e1	3.5e1	15	1.7e3	1.5e3	1.9e3	1.7e3
1	15	1.3e2	1.2e2	1.3e2	1.3e2	15	4.1e3	3.7e3	4.6e3	4.1e3
1e−1	15	2.1e2	2.1e2	2.2e2	2.1e2	15	5.0e3	4.5e3	5.5e3	5.0e3
1e−3	15	3.0e2	2.9e2	3.1e2	3.0e2	15	5.7e3	5.1e3	6.2e3	5.7e3
1e−5	15	3.4e2	3.2e2	3.5e2	3.4e2	15	6.1e3	5.5e3	6.7e3	6.1e3
1e−8	15	3.7e2	3.6e2	3.8e2	3.7e2	15	6.3e3	5.6e3	6.9e3	6.3e3

	f_{10} in 5-D, N=15, mFE=50179					**f_{10} in 20-D**, N=15, mFE=81192				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	6	9.7e4	6.7e4	1.6e5	4.0e4	0	*72e+2*	*42e+2*	*11e+3*	4.0e4
1	0	*17e+0*	*39e-1*	*74e+1*	2.8e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{11} in 5-D, N=15, mFE=50203					**f_{11} in 20-D**, N=15, mFE=81248				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.2e4	7.5e3	1.6e4	1.2e4	0	*62e+0*	*35e+0*	*99e+0*	2.8e4
1	6	9.3e4	6.2e4	1.6e5	3.3e4	.	.	.	.	.
1e−1	0	*13e-1*	*36e-2*	*41e-1*	2.0e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{12} in 5-D, N=15, mFE=50078					**f_{12} in 20-D**, N=15, mFE=80828				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.1e2	1.0e2	1.1e2	1.1e2	15	1.1e3	9.2e2	1.3e3	1.1e3
1	15	4.8e3	1.8e3	8.2e3	4.8e3	15	1.6e4	7.5e3	2.5e4	1.6e4
1e−1	15	6.2e3	3.3e3	9.5e3	6.2e3	13	3.3e4	2.3e4	4.5e4	3.2e4
1e−3	14	1.0e4	5.7e3	1.5e4	1.0e4	6	1.8e5	1.2e5	2.9e5	6.5e4
1e−5	10	3.4e4	2.2e4	5.0e4	2.2e4	1	1.2e6	5.6e5	>1e6	8.1e4
1e−8	6	8.4e4	5.3e4	1.5e5	2.9e4	0	*16e-4*	*33e-6*	*16e-2*	5.0e4

	f_{13} in 5-D, N=15, mFE=50037					**f_{13} in 20-D**, N=15, mFE=80543				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	5.4e3	4.0e3	6.9e3	5.4e3	14	2.2e4	1.5e4	3.0e4	2.2e4
1	11	4.2e4	3.4e4	5.3e4	3.4e4	11	7.5e4	5.7e4	9.9e4	5.2e4
1e−1	5	1.2e5	8.0e4	2.1e5	4.8e4	6	1.7e5	1.2e5	2.8e5	7.1e4
1e−3	1	7.2e5	3.5e5	>7e5	5.0e4	0	*60e-2*	*15e-3*	*71e-1*	4.5e4
1e−5	0	*21e-2*	*10e-3*	*66e-1*	2.0e4	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{14} in 5-D, N=15, mFE=50120					**f_{14} in 20-D**, N=15, mFE=81243				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.4e1	8.0e0	1.9e1	1.4e1	15	7.5e1	6.5e1	8.5e1	7.5e1
1	15	1.1e2	7.8e1	1.5e2	1.1e2	15	5.0e2	4.4e2	5.5e2	5.0e2
1e−1	15	1.5e2	1.2e2	1.9e2	1.5e2	15	1.0e3	9.6e2	1.1e3	1.0e3
1e−3	15	3.8e2	3.5e2	4.2e2	3.8e2	15	3.0e3	2.8e3	3.2e3	3.0e3
1e−5	8	5.7e4	3.8e4	8.9e4	2.8e4	0	*15e-5*	*12e-5*	*20e-5*	2.8e4
1e−8	0	*10e-6*	*35e-7*	*26e-6*	2.8e4	.	.	.	.	.

	f_{15} in 5-D, N=15, mFE=50598					**f_{15} in 20-D**, N=15, mFE=81125				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	2.1e3	1.6e3	2.5e3	2.1e3	0	*10e+1*	*72e+0*	*21e+1*	4.0e4
1	3	2.4e5	1.4e5	7.1e5	5.0e4	.	.	.	.	.
1e−1	1	7.4e5	3.7e5	>7e5	5.0e4	.	.	.	.	.
1e−3	1	7.4e5	3.7e5	>7e5	5.0e4	.	.	.	.	.
1e−5	1	7.5e5	3.7e5	>7e5	5.0e4	.	.	.	.	.
1e−8	0	*20e-1*	*46e-2*	*40e-1*	2.2e4	.	.	.	.	.

	f_{16} in 5-D, N=15, mFE=50351					**f_{16} in 20-D**, N=15, mFE=81115				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	2.2e2	1.5e2	3.1e2	2.2e2	15	1.5e4	1.0e4	2.1e4	1.5e4
1	15	1.1e4	7.3e3	1.5e4	1.1e4	0	*75e-1*	*62e-1*	*93e-1*	2.8e4
1e−1	2	3.5e5	1.8e5	>7e5	5.0e4	.	.	.	.	.
1e−3	0	*30e-2*	*69e-3*	*70e-2*	2.8e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{17} in 5-D, N=15, mFE=50609					**f_{17} in 20-D**, N=15, mFE=81341				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.0e1	7.4e0	1.3e1	1.0e1	15	6.3e1	4.5e1	8.1e1	6.3e1
1	15	5.2e3	4.0e3	6.5e3	5.2e3	0	*42e-1*	*35e-1*	*54e-1*	2.0e4
1e−1	8	5.6e4	4.0e4	8.4e4	3.3e4	.	.	.	.	.
1e−3	0	*87e-3*	*72e-3*	*34e-2*	7.9e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{18} in 5-D, N=15, mFE=50523					**f_{18} in 20-D**, N=15, mFE=81198				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.9e3	1.2e3	2.6e3	1.9e3	0	*17e+0*	*14e+0*	*23e+0*	4.0e4
1	9	5.8e4	4.5e4	8.1e4	3.7e4	.	.	.	.	.
1e−1	0	*75e-2*	*23e-2*	*14e-1*	2.5e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{19} in 5-D, N=15, mFE=50827					**f_{19} in 20-D**, N=15, mFE=80858				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1e−1	15	2.4e2	2.2e2	2.7e2	2.4e2	3	3.4e5	2.1e5	1.0e6	8.1e4
1e−3	0	*16e-3*	*16e-3*	*16e-3*	1.0e4	0	*25e-2*	*16e-3*	*25e-2*	1.0e0
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{20} in 5-D, N=15, mFE=50402					**f_{20} in 20-D**, N=15, mFE=80342				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	4.4e1	3.7e1	5.1e1	4.4e1	15	3.8e2	3.7e2	3.9e2	3.8e2
1	15	8.5e2	4.7e2	1.2e3	8.5e2	2	5.7e5	2.8e5	>1e6	5.6e4
1e−1	2	3.5e5	1.7e5	>7e5	3.3e4	0	*12e-1*	*10e-1*	*14e-1*	5.6e4
1e−3	2	3.5e5	1.7e5	>7e5	3.3e4	.	.	.	.	.
1e−5	2	3.5e5	1.7e5	>7e5	3.3e4	.	.	.	.	.
1e−8	2	3.5e5	1.7e5	>7e5	3.3e4	.	.	.	.	.

	f_{21} in 5-D, N=15, mFE=50061					**f_{21} in 20-D**, N=15, mFE=80340				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	4.2e1	3.4e1	5.1e1	4.2e1	15	1.4e4	8.3e3	2.0e4	1.4e4
1	15	4.6e3	2.3e3	6.9e3	4.6e3	5	2.1e5	1.4e5	3.8e5	6.5e4
1e−1	14	8.5e3	4.1e3	1.3e4	8.3e3	3	3.6e5	2.1e5	1.1e6	5.4e4
1e−3	14	8.6e3	4.3e3	1.3e4	8.4e3	3	3.6e5	2.1e5	1.1e6	5.5e4
1e−5	14	8.7e3	4.4e3	1.3e4	8.5e3	3	3.7e5	2.1e5	1.1e6	5.7e4
1e−8	14	9.0e3	4.9e3	1.4e4	8.8e3	2	5.8e5	2.8e5	>1e6	5.9e4

	f_{22} in 5-D, N=15, mFE=50081					**f_{22} in 20-D**, N=15, mFE=80286				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	7.4e1	6.2e1	8.5e1	7.4e1	15	7.7e3	3.0e3	1.2e4	7.7e3
1	15	4.2e2	3.0e2	5.5e2	4.2e2	7	1.1e5	7.9e4	1.8e5	5.6e4
1e−1	15	1.1e4	7.6e3	1.4e4	1.1e4	1	1.2e6	5.7e5	>1e6	8.0e4
1e−3	15	1.1e4	7.6e3	1.4e4	1.1e4	1	1.2e6	5.7e5	>1e6	8.0e4
1e−5	15	1.1e4	7.8e3	1.4e4	1.1e4	0	*20e-1*	*69e-2*	*73e-1*	4.0e4
1e−8	14	2.0e4	1.5e4	2.5e4	2.0e4	.	.	.	.	.

	f_{23} in 5-D, N=15, mFE=50687					**f_{23} in 20-D**, N=15, mFE=81366				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.0e1	7.7e0	1.3e1	1.0e1	15	4.2e0	3.0e0	5.5e0	4.2e0
1	15	1.2e3	9.0e2	1.6e3	1.2e3	5	2.0e5	1.4e5	3.5e5	8.1e4
1e−1	1	7.3e5	3.5e5	>7e5	5.1e4	0	*11e-1*	*55e-2*	*13e-1*	4.5e4
1e−3	0	*16e-2*	*12e-2*	*28e-2*	2.0e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{24} in 5-D, N=15, mFE=50632					**f_{24} in 20-D**, N=15, mFE=80986				
Δf	#	ERT	10%	90%	RT$_{\text{succ}}$	#	ERT	10%	90%	RT$_{\text{succ}}$
10	15	1.1e4	8.2e3	1.4e4	1.1e4	0	*10e+1*	*91e+0*	*12e+1*	2.0e4
1	1	7.5e5	3.7e5	>8e5	5.0e4	.	.	.	.	.
1e−1	0	*37e-1*	*14e-1*	*46e-1*	4.0e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 1: Shown are, for a given target difference to the optimal function value Δf: the number of successful trials (#); the expected running time to surpass $f_{\text{opt}} + \Delta f$ (ERT, see Figure 1); 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 (RT$_{\text{succ}}$). If $f_{\text{opt}} + \Delta f$ 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 1 for the names of functions.

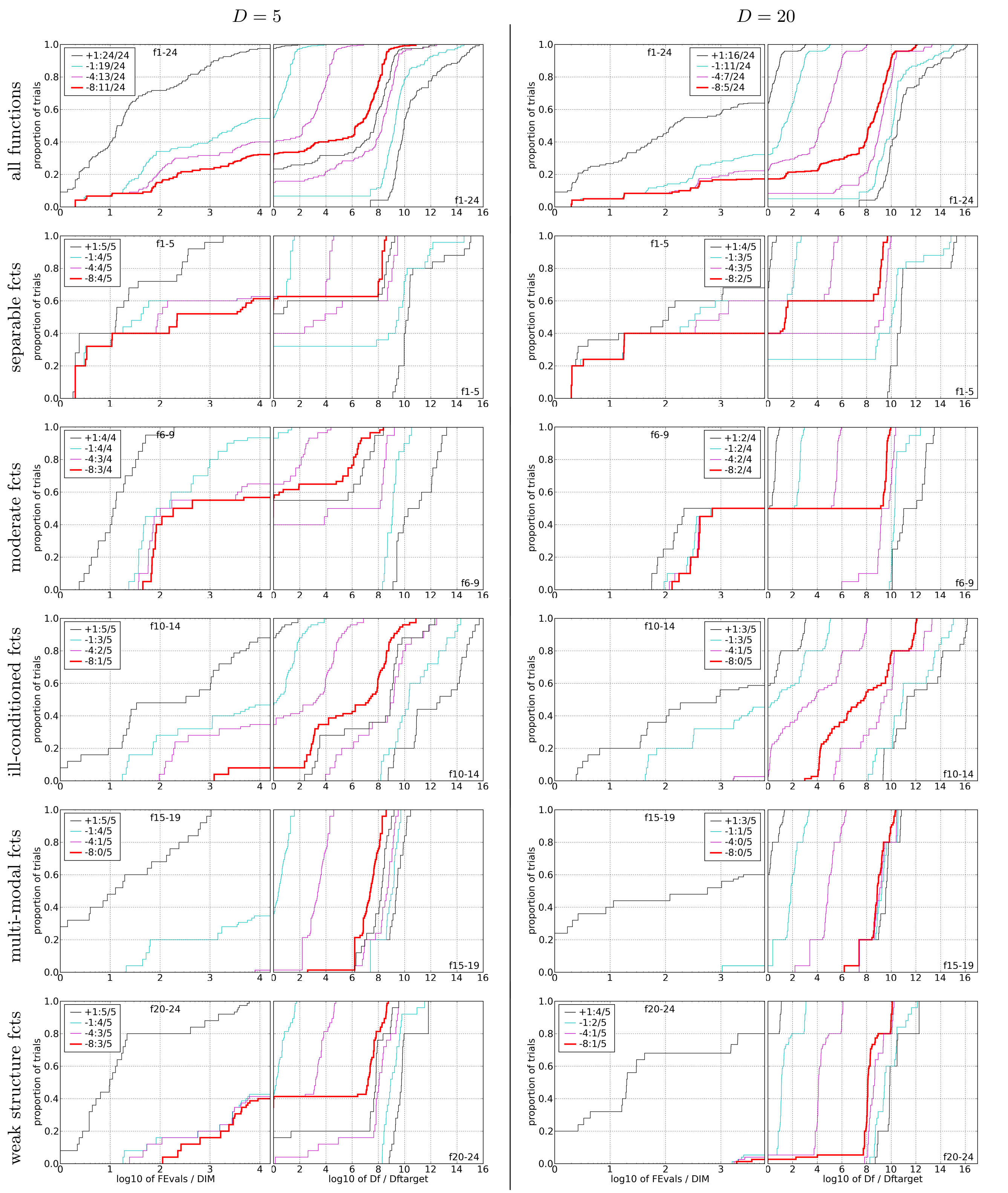

Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left) or Δf. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\rm opt} + \Delta f$ with $\Delta f = 10^k$, 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.