Benchmarking of MCS on the Noisy 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 noisy 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.

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 $s \in \{1, \dots, s_{\max}\}$ 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. Since it does not make sense to apply local search to a noisy function, we use MCS without local search, where the points and function values belonging to boxes of level $s_{\max}$ are put into the so-called shopping basket (containing 'useful' points) without processing them further.

The algorithm starts with a so-called initialization procedure producing an initial set of boxes. For each coordinate $i = 1, \dots, n$, 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 $l_i \in \{1, \dots, L_i\}$ point to the initial point x^0, i.e., $x_i^0 = x_i^{l_i}$. The values x_i^j, $j = 1, \dots, L_i$, l_i, and L_i, $i = 1, \dots, n$, 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, and the initialization list. The limit on function calls in each local search is set to 0 (no 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, $u = (-5, \dots, -5)^T$ and $v = (5, \dots, 5)^T$. 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 $x_i^3 = v_i$, $i = 1, \dots, n$. Another initialization list for finite 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

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

is possible to use a self-defined initialization list.

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 $nf_{\max}$ 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.

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 Xeon 3.4 GHz under SuSE Linux with MATLAB 7.3.0.298, where the benchmarking tests were run. The results were 22, 7.4, 4.8, 4.0, 19, and 6.2 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 benchmarks functions given in [1, 3] are presented in Figures 1 and 2 and in Tables 1 and 2.

6. REFERENCES

[1] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noisy functions. Technical Report 2009/21, 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: Noisy functions definitions. Technical Report RR-6869, 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.

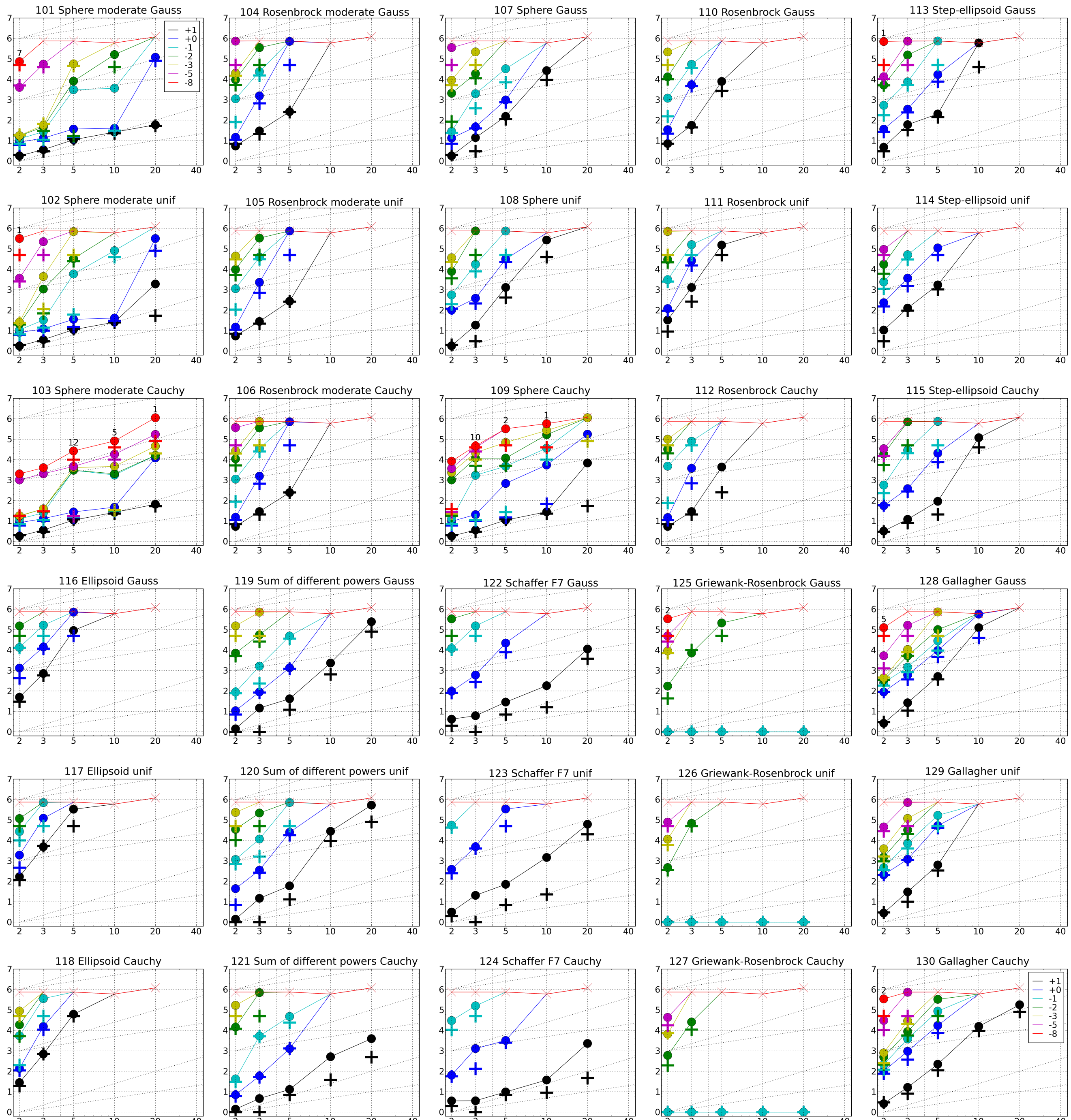

Figure 1: Expected Running Time (ERT, ●) to reach $f_{\mathrm{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_{101} and f_{130}) versus dimension in 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 (×) 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.

	f101 in 5-D, N=15, mFE=50009					**f101 in 20-D**, N=15, mFE=80037				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	1.1e1	1.1e1	1.1e1	1.1e1	15	5.9e1	5.5e1	6.4e1	5.9e1
1	15	3.7e1	2.2e1	5.2e1	3.7e1	6	1.2e5	8.0e4	2.0e5	5.3e4
1e−1	15	3.1e3	1.7e3	4.4e3	3.1e3	0	*11e−1*	*21e−2*	*21e−1*	1.1e4
1e−3	8	5.7e4	4.8e4	7.4e4	4.5e4	.	.	.	.	.
1e−5	0	*82e−5*	*22e−5*	*31e−4*	1.0e4	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f103 in 5-D, N=15, mFE=50006					**f103 in 20-D**, N=15, mFE=80024				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	1.1e1	1.1e1	1.1e1	1.1e1	15	6.5e1	5.9e1	7.2e1	6.5e1
1	15	2.8e1	1.9e1	3.6e1	2.8e1	15	1.2e4	9.4e3	1.5e4	1.2e4
1e−1	15	3.0e3	1.7e3	4.4e3	3.0e3	15	1.6e4	1.3e4	1.9e4	1.6e4
1e−3	15	4.0e3	2.7e3	5.4e3	4.0e3	11	4.6e4	2.9e4	6.7e4	2.7e4
1e−5	15	4.7e3	3.4e3	6.0e3	4.7e3	5	1.7e5	1.1e5	3.3e5	5.2e4
1e−8	12	2.7e4	1.9e4	3.5e4	2.7e4	1	1.1e6	5.2e5	>1e6	8.0e4

	f105 in 5-D, N=15, mFE=50018					**f105 in 20-D**, N=15, mFE=80026				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	2.7e2	2.4e2	3.1e2	2.7e2	0	*19e+1*	*15e+1*	*11e+2*	6.3e3
1	1	7.4e5	3.7e5	>7e5	5.0e4	.	.	.	.	.
1e−1	0	*32e−1*	*11e−1*	*47e−1*	3.2e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f107 in 5-D, N=15, mFE=50025					**f107 in 20-D**, N=15, mFE=80038				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	1.5e2	1.1e2	2.0e2	1.5e2	0	*64e+0*	*45e+0*	*79e+0*	1.8e4
1	15	9.6e2	7.3e2	1.2e3	9.6e2	.	.	.	.	.
1e−1	10	3.3e4	2.1e4	4.8e4	2.1e4	.	.	.	.	.
1e−3	0	*81e−3*	*34e−3*	*16e−2*	6.3e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f109 in 5-D, N=15, mFE=50009					**f109 in 20-D**, N=15, mFE=80030				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	1.1e1	1.1e1	1.1e1	1.1e1	15	6.8e3	1.4e3	1.3e4	6.8e3
1	15	6.9e2	1.9e1	1.4e3	6.9e2	5	1.8e5	1.1e5	3.2e5	5.6e4
1e−1	15	4.7e3	2.0e3	7.6e3	4.7e3	1	1.1e6	5.2e5	>1e6	8.0e4
1e−3	7	6.9e4	4.6e4	1.1e5	3.1e4	1	1.1e6	5.2e5	>1e6	8.0e4
1e−5	2	3.3e5	1.6e5	>7e5	2.5e4	0	*13e−1*	*56e−2*	*41e−1*	2.0e4
1e−8	2	3.3e5	1.6e5	>7e5	2.5e4	.	.	.	.	.

	f111 in 5-D, N=15, mFE=50020					**f111 in 20-D**, N=15, mFE=80041				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	4	1.5e5	9.1e4	3.4e5	3.0e4	0	*21e+3*	*83e+2*	*49e+3*	1.8e4
1	0	*12e+0*	*60e−1*	*27e+0*	8.9e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f113 in 5-D, N=15, mFE=50020					**f113 in 20-D**, N=15, mFE=80039				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	2.0e2	1.4e2	2.7e2	2.0e2	0	*22e+1*	*10e+1*	*56e+1*	2.2e4
1	13	1.7e4	1.2e4	2.3e4	1.6e4	.	.	.	.	.
1e−1	1	7.4e5	3.7e5	>7e5	5.0e4	.	.	.	.	.
1e−3	0	*48e−2*	*21e−2*	*12e−1*	2.5e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f115 in 5-D, N=15, mFE=50019					**f115 in 20-D**, N=15, mFE=80041				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	9.2e1	4.7e1	1.4e2	9.2e1	0	*72e+0*	*47e+0*	*13e+1*	1.6e4
1	12	2.1e4	1.3e4	3.0e4	1.7e4	.	.	.	.	.
1e−1	1	7.4e5	3.7e5	>7e5	5.0e4	.	.	.	.	.
1e−3	0	*83e−2*	*29e−2*	*12e−1*	8.9e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f117 in 5-D, N=15, mFE=50025					**f117 in 20-D**, N=15, mFE=80045				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	2	3.3e5	1.8e5	>7e5	5.0e4	0	*20e+3*	*60e+2*	*48e+3*	1.0e4
1	0	*36e+0*	*92e−1*	*57e+0*	1.6e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f119 in 5-D, N=15, mFE=50020					**f119 in 20-D**, N=15, mFE=80041				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	4.1e1	1.9e1	6.4e1	4.1e1	4	2.4e5	1.5e5	5.1e5	6.9e4
1	15	1.3e3	1.0e3	1.6e3	1.3e3	0	*14e+0*	*73e−1*	*19e+0*	1.4e4
1e−1	9	4.8e4	3.5e4	6.9e4	3.2e4	.	.	.	.	.
1e−3	0	*75e−3*	*43e−3*	*30e−2*	1.8e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f102 in 5-D, N=15, mFE=50014					**f102 in 20-D**, N=15, mFE=80033				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	1.1e1	1.1e1	1.2e1	1.1e1	15	1.9e3	3.3e2	3.6e3	1.9e3
1	15	3.5e1	2.3e1	4.9e1	3.5e1	3	3.2e5	1.9e5	9.6e5	8.0e4
1e−1	15	5.9e3	2.8e3	9.2e3	5.9e3	0	*25e−1*	*56e−2*	*57e−1*	6.3e3
1e−3	1	7.0e5	3.3e5	>7e5	2.0e1	.	.	.	.	.
1e−5	0	*58e−4*	*13e−4*	*21e−3*	2.8e4	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f104 in 5-D, N=15, mFE=50006					**f104 in 20-D**, N=15, mFE=80032				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	2.6e2	2.3e2	2.9e2	2.6e2	0	*20e+1*	*16e+1*	*88e+1*	1.0e4
1	1	7.1e5	3.4e5	>7e5	5.0e4	.	.	.	.	.
1e−1	0	*23e−1*	*13e−1*	*42e−1*	2.5e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f106 in 5-D, N=15, mFE=50008					**f106 in 20-D**, N=15, mFE=80023				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	2.4e2	2.2e2	2.7e2	2.4e2	0	*20e+1*	*13e+1*	*12e+2*	1.4e4
1	1	7.1e5	3.4e5	>7e5	5.0e4	.	.	.	.	.
1e−1	0	*27e−1*	*10e−1*	*44e−1*	1.6e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f108 in 5-D, N=15, mFE=50022					**f108 in 20-D**, N=15, mFE=80041				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	1.3e3	8.0e2	1.8e3	1.3e3	0	*69e+0*	*52e+0*	*84e+0*	1.8e4
1	12	3.1e4	2.3e4	4.1e4	2.6e4	.	.	.	.	.
1e−1	1	7.4e5	3.6e5	>7e5	5.0e4	.	.	.	.	.
1e−3	0	*54e−2*	*19e−2*	*12e−1*	3.2e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f110 in 5-D, N=15, mFE=50021					**f110 in 20-D**, N=15, mFE=80040				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	7.9e3	4.4e3	1.2e4	7.9e3	0	*14e+3*	*57e+2*	*36e+3*	1.6e4
1	0	*54e−1*	*31e−1*	*92e−1*	4.0e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f112 in 5-D, N=15, mFE=50008					**f112 in 20-D**, N=15, mFE=80024				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	14	4.4e3	2.9e2	8.6e3	4.4e3	0	*20e+1*	*16e+1*	*12e+2*	1.1e4
1	0	*37e−1*	*13e−1*	*57e−1*	1.3e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f114 in 5-D, N=15, mFE=50032					**f114 in 20-D**, N=15, mFE=80042				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	1.7e3	9.6e2	2.5e3	1.7e3	0	*30e+1*	*18e+1*	*77e+1*	2.0e4
1	5	1.1e5	7.5e4	2.0e5	4.2e4	.	.	.	.	.
1e−1	0	*13e−1*	*35e−2*	*29e−1*	8.9e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f116 in 5-D, N=15, mFE=50025					**f116 in 20-D**, N=15, mFE=80037				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	6	9.1e4	6.1e4	1.6e5	3.5e4	0	*12e+3*	*64e+2*	*16e+3*	1.8e4
1	1	7.1e5	3.3e5	>7e5	5.0e4	.	.	.	.	.
1e−1	0	*12e+0*	*35e−1*	*26e+0*	7.9e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f118 in 5-D, N=15, mFE=50009					**f118 in 20-D**, N=15, mFE=80024				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	7	6.1e4	4.3e4	9.5e4	3.7e4	0	*26e+2*	*21e+2*	*50e+2*	1.4e4
1	0	*13e+0*	*23e−1*	*24e+0*	7.1e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f120 in 5-D, N=15, mFE=50025					**f120 in 20-D**, N=15, mFE=80043				
Δf	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	15	6.0e1	2.0e1	1.0e2	6.0e1	2	5.3e5	2.8e5	>1e6	8.0e4
1	13	2.5e4	1.8e4	3.3e4	2.1e4	0	*15e+0*	*97e−1*	*21e+0*	1.1e4
1e−1	1	7.2e5	3.4e5	>7e5	5.0e4	.	.	.	.	.
1e−3	0	*48e−2*	*13e−2*	*11e−1*	2.8e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 1: Shown are, for functions f_{101}-f_{120} and 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 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 ($\mathbf{RT_{succ}}$). 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 1 for the names of functions.

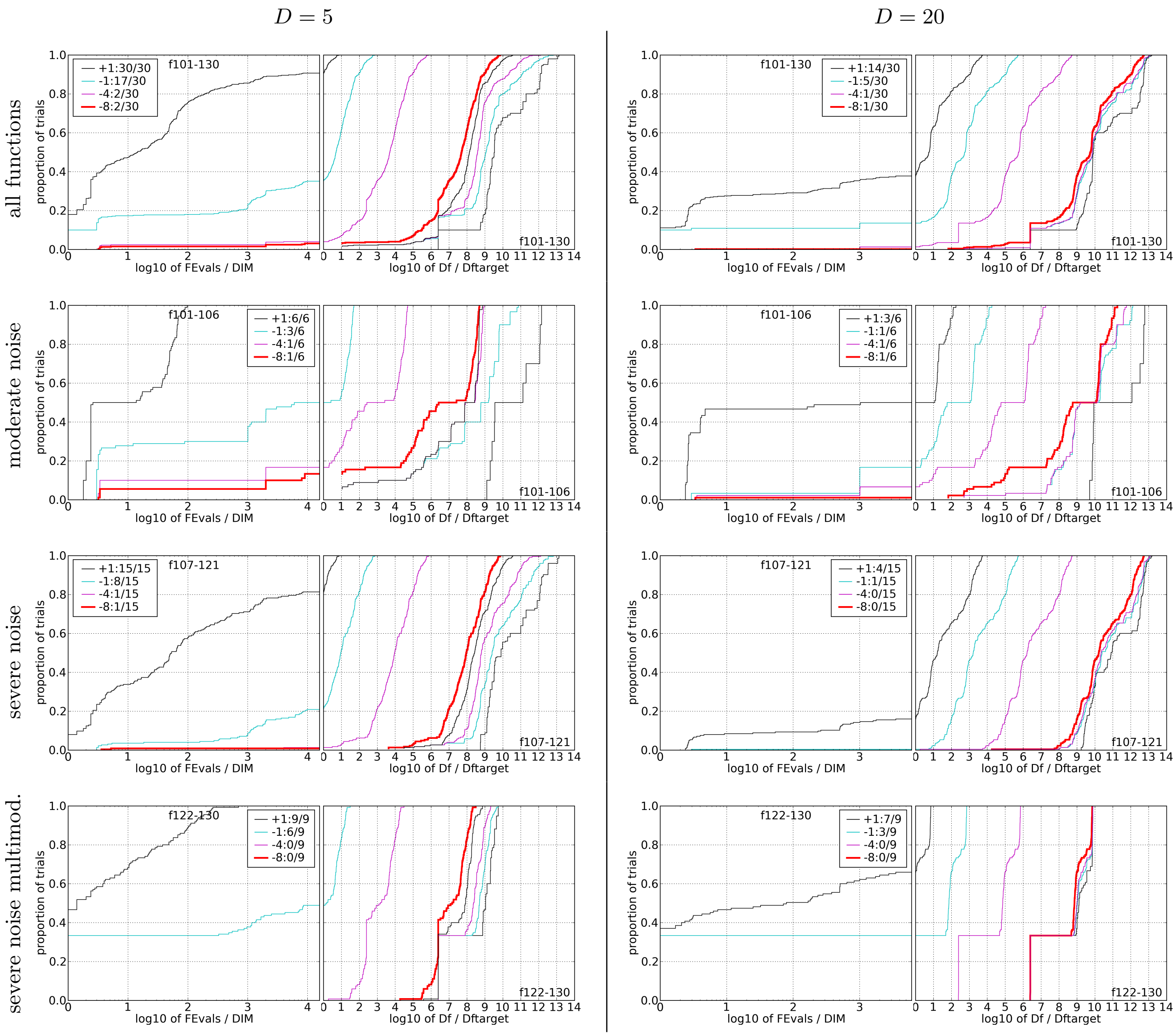

Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left) or Δf. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\mathrm{opt}} + \Delta f$ with $\Delta f =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

	***f*121 in 5-D**, N=15, mFE=50011					***f*121 in 20-D**, N=15, mFE=80026				
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.3e1	8.0e0	1.8e1	1.3e1	15	3.9e3	2.3e3	5.5e3	3.9e3
1	15	1.3e3	9.8e2	1.6e3	1.3e3	0	*60e−1*	*39e−1*	*87e−1*	1.0e4
1e−1	9	4.8e4	3.5e4	6.8e4	3.1e4	.	.	.	.	.
1e−3	0	*88e−3*	*45e−3*	*20e−2*	8.9e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	***f*122 in 5-D**, N=15, mFE=50026					***f*122 in 20-D**, N=15, mFE=80037				
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.8e1	1.3e1	4.5e1	2.8e1	15	1.1e4	4.3e3	1.9e4	1.1e4
1	13	2.2e4	1.5e4	3.1e4	1.8e4	0	*77e−1*	*62e−1*	*90e−1*	2.0e4
1e−1	0	*54e−2*	*36e−2*	*11e−1*	8.9e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	***f*123 in 5-D**, N=15, mFE=50024					***f*123 in 20-D**, N=15, mFE=80043				
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	7.1e1	1.8e1	1.3e2	7.1e1	9	6.2e4	4.0e4	9.1e4	4.0e4
1	2	3.4e5	1.6e5	>7e5	2.6e4	0	*87e−1*	*66e−1*	*16e+0*	2.2e4
1e−1	0	*16e−1*	*93e−2*	*18e−1*	1.6e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	***f*124 in 5-D**, N=15, mFE=50016					***f*124 in 20-D**, N=15, mFE=80033				
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	9.7e0	7.1e0	1.2e1	9.7e0	15	2.3e3	7.7e2	4.0e3	2.3e3
1	15	3.2e3	2.2e3	4.2e3	3.2e3	0	*66e−1*	*49e−1*	*73e−1*	2.0e4
1e−1	0	*55e−2*	*27e−2*	*84e−2*	1.3e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	***f*125 in 5-D**, N=15, mFE=50014					***f*125 in 20-D**, N=15, mFE=80039				
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{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	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1e−3	0	*15e−3*	*63e−4*	*25e−3*	1.1e4	0	*25e−3*	*25e−3*	*25e−3*	1.0e0
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	***f*126 in 5-D**, N=15, mFE=50021					***f*126 in 20-D**, N=15, mFE=80038				
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{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	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1e−3	0	*25e−3*	*24e−3*	*25e−3*	1.0e0	0	*25e−3*	*25e−3*	*25e−3*	1.0e0
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	***f*127 in 5-D**, N=15, mFE=50025					***f*127 in 20-D**, N=15, mFE=80039				
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{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	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1e−3	0	*25e−3*	*20e−3*	*25e−3*	1.0e0	0	*25e−3*	*25e−3*	*25e−3*	1.0e0
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	***f*128 in 5-D**, N=15, mFE=50024					***f*128 in 20-D**, N=15, mFE=80040				
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	5.1e2	3.7e2	6.4e2	5.1e2	0	*66e+0*	*60e+0*	*71e+0*	2.8e4
1	15	9.6e3	5.5e3	1.4e4	9.6e3	.	.	.	.	.
1e−1	11	2.8e4	2.0e4	3.7e4	2.6e4	.	.	.	.	.
1e−3	1	7.4e5	3.6e5	>7e5	5.0e4	.	.	.	.	.
1e−5	0	*49e−3*	*31e−4*	*18e−2*	2.2e4	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	***f*129 in 5-D**, N=15, mFE=50021					***f*129 in 20-D**, N=15, mFE=80038				
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	6.3e2	3.7e2	9.3e2	6.3e2	0	*67e+0*	*58e+0*	*71e+0*	2.0e4
1	9	5.2e4	3.6e4	8.0e4	2.6e4	.	.	.	.	.
1e−1	4	1.7e5	1.0e5	3.5e5	4.5e4	.	.	.	.	.
1e−3	0	*64e−2*	*52e−3*	*21e−1*	1.8e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	***f*130 in 5-D**, N=15, mFE=50011					***f*130 in 20-D**, N=15, mFE=80033				
Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.2e2	1.4e2	3.0e2	2.2e2	5	1.8e5	1.2e5	3.3e5	7.2e4
1	13	1.7e4	1.1e4	2.4e4	1.7e4	0	*12e+0*	*73e−1*	*35e+0*	2.8e4
1e−1	6	8.7e4	5.7e4	1.5e5	3.4e4	.	.	.	.	.
1e−3	0	*35e−2*	*32e−4*	*11e−1*	7.9e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 2: Shown are, for functions f_{121}-f_{130} and 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 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 ($\mathbf{RT_{succ}}$). 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 1 for the names of functions.