Benchmarking of SNOBFIT 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 SNOBFIT algorithm for noisy 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

The algorithm SNOBFIT (stable noisy optimization by branch and fit) [4] for bound-constrained global optimization of noisy functions combines global and local search by branching (i.e., splitting the search space $[u, v]$ into smaller boxes) and local fits. Based on function values already available, the algorithm builds internally around each point local models of the function to minimize, and returns in each step a number of points whose evaluation is likely to improve these models or is expected to give better function values. Surrogate functions are not interpolated but fitted to a stochastic (linear or quadratic) model to take noisy function values into account.

2. ALGORITHM PRESENTATION

The optimization proceeds through a number of calls to SNOBFIT producing a user-specified number of new recommended evaluation points. SNOBFIT generates points of different character belonging to five classes. Class 1 contains at most one point, determined from the local quadratic model around the best point. Points of classes 2 and 3 are alternative good points selected with a view to their expected function value. The points in class 4 are points in regions unexplored thus far, i.e., they are generated in large subboxes of the current partition. Points of class 5 are only produced if the algorithm does not manage to reach the desired number of points by generating points of classes 1 to 4, for example, when there are not enough points yet available to build local quadratic models, and they are chosen from a set of random points such that their distances from the points already in the list are maximal.

In the so-called initial call, a 'resolution vector' $\Delta x > 0$ is needed as an additional input. It is assumed that the ith coordinate is measured in units Δx_i. The algorithm only suggests evaluation points whose ith coordinate is an integral multiple of Δx_i. In each call to SNOBFIT, a list x^j, $j = 1, \dots, J$ of points, their corresponding function values f_j, the uncertainties Δf_j of the function values, a natural number $n_{\rm req}$, two vectors u' and v', and a real number $p \in [0, 1]$ (the desired fraction of points of class 4 among the points of classes 2 to 4, i.e., p controls whether local or global search should be emphasized) are fed into the program. The program then returns $n_{\rm req}$ (occasionally fewer) suggested evaluation points in the box $[u', v']$, their classes, and the model function values at these points. The idea of the algorithm is that these points and their function values are used as input for the next call to SNOBFIT.

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

3. EXPERIMENTAL PROCEDURE

In our experiments, we always set $u' = u = (-5, \dots, -5)^T$ and $v' = v = (5, \dots, 5)^T$ (i.e., the points are to be generated in the whole search space), $\Delta x = 10^{-8}(v - u)$ and $p = 0.1$, and we generate $n_{\rm req} = n + 6$ points in each call to SNOBFIT, where n is the dimension of the problem. These values are considered to be meaningful default values for the algorithm. If f is the function value at a point, the uncertainty Δf is set to $0.03f$ for the functions f_{101} to f_{106} (moderate noise) and $0.3f$ for the functions f_{107} to f_{130} (severe noise). We start each trial with an initial call with $n + 6$ points drawn uniformly from $[u, v]$ as input, and we proceed by so-called continuation calls to SNOBFIT till 1000 function evaluations (including the ones made in the initial call) have been reached. Afterwards we repeat this procedure (at most) four times, but instead of sampling $n + 6$ points for

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

the initial calls, we only sample $n+5$ points and in addition keep the best point from the previous 'iteration'. I.e., each trial consists of at most 5 attempts to solve the problem with the SNOBFIT algorithm, and only the best point from the previous attempt is reused. After each call to SNOBFIT, it is checked whether the target function value f_{target} has been reached, and in that case the trial is terminated. So at most 5000 function calls (possibly a few more since reaching the number of permitted function values is only checked after each call to SNOBFIT) are made in each trial. Three trials are made for the 5 function instances of each function.

SNOBFIT uses the program MINQ for minimizing a quadratic model, which can be downloaded from `http://www.mat.univie.ac.at/~neum/software/minq/`. Since the iteration limit in MINQ was frequently exceeded in higher dimensions, we changed the line `maxit=3*n` in `minq.m` to `maxit=10*n`.

4. CPU TIMING EXPERIMENT

For the timing experiment according to [2], the experimental procedure described above was run on f_8 with at most 100 function evaluations in each SNOBFIT 'iteration' and restarted until at least 30 seconds had passed. The uncertainties were set to $\Delta f = \sqrt{\varepsilon} f$ (where ε is the machine precision) since f_8 is a noiseless function. 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 8.3, 8.5, 8.8, 9.1, 10, and 7.9 times 10^{-1} 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. SNOBFIT - stable noisy optimization by branch and fit. *ACM Trans. Math. Software*, 35(2), 2008.

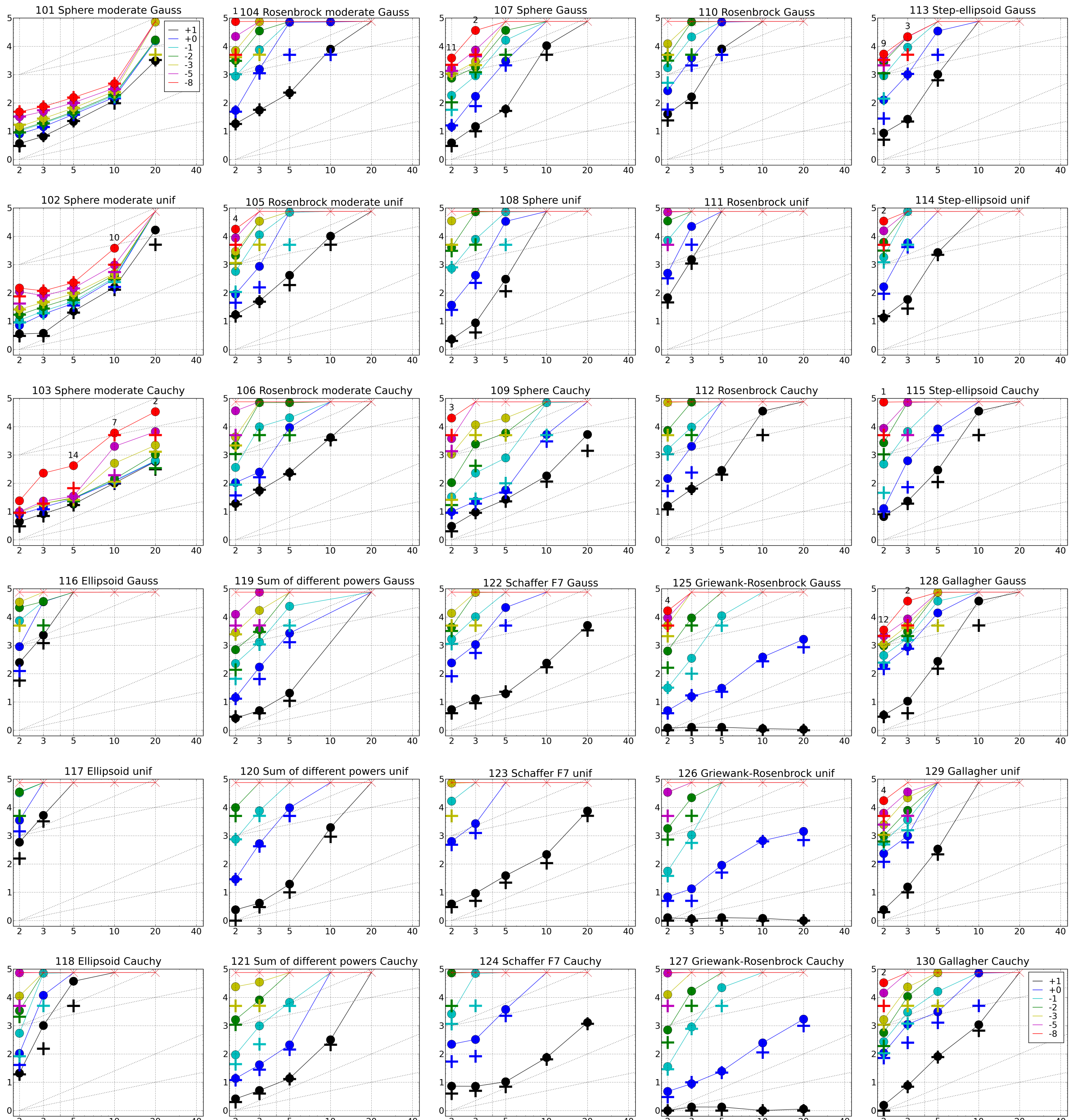

Figure 1: Expected Running Time (ERT, ●) to reach $f_{\rm opt} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_{101} and f_{130}) versus dimension in log-log presentation. The ERT(Δf) equals to #FEs(Δf) divided by the number of successful trials, where a trial is successful if $f_{\rm opt} + \Delta f$ was surpassed during the trial. The #FEs(Δf) 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 (×) 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.

Δf	**f101 in 5-D**, N=15, mFE=198					**f101 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	2.2e1	1.7e1	2.6e1	2.2e1	13	3.1e3	2.3e3	4.0e3	2.5e3
1	15	3.9e1	3.5e1	4.3e1	3.9e1	4	1.5e4	9.6e3	3.4e4	4.0e3
1e−1	15	4.4e1	4.0e1	4.8e1	4.4e1	4	1.5e4	9.7e3	3.3e4	4.0e3
1e−3	15	6.6e1	6.1e1	7.0e1	6.6e1	1	7.2e4	3.4e4	>7e4	5.1e3
1e−5	15	1.0e2	9.7e1	1.1e2	1.0e2	0	*37e−1*	*37e−4*	*12e+0*	4.5e3
1e−8	15	1.6e2	1.5e2	1.6e2	1.6e2	.	.	.	.	.

Δf	**f102 in 5-D**, N=15, mFE=275					**f102 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	2.2e1	1.8e1	2.7e1	2.2e1	4	1.7e4	1.0e4	3.5e4	3.8e3
1	15	4.3e1	3.8e1	4.7e1	4.3e1	0	*18e+0*	*52e−1*	*32e+0*	4.0e3
1e−1	15	5.0e1	4.5e1	5.5e1	5.0e1	.	.	.	.	.
1e−3	15	9.4e1	8.7e1	1.0e2	9.4e1	.	.	.	.	.
1e−5	15	1.5e2	1.4e2	1.5e2	1.5e2	.	.	.	.	.
1e−8	15	2.3e2	2.2e2	2.4e2	2.3e2	.	.	.	.	.

Δf	**f103 in 5-D**, N=15, mFE=5001					**f103 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	1.8e1	1.5e1	2.1e1	1.8e1	15	5.8e2	4.4e2	7.3e2	5.8e2
1	15	2.9e1	2.6e1	3.1e1	2.9e1	15	6.0e2	4.6e2	7.6e2	6.0e2
1e−1	15	3.0e1	2.7e1	3.3e1	3.0e1	15	6.3e2	4.8e2	7.8e2	6.3e2
1e−3	15	3.1e1	2.8e1	3.5e1	3.1e1	12	2.2e3	1.4e3	3.2e3	1.6e3
1e−5	15	3.5e1	3.1e1	3.8e1	3.5e1	7	6.6e3	4.1e3	1.2e4	2.0e3
1e−8	14	4.2e2	6.0e1	8.4e2	4.1e2	2	3.4e4	1.8e4	>7e4	5.1e3

Δf	**f104 in 5-D**, N=15, mFE=5001					**f104 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	2.2e2	2.0e2	2.5e2	2.2e2	0	*63e+2*	*19e+2*	*10e+3*	4.5e3
1	1	7.0e4	3.3e4	>7e4	3.1e2	.	.	.	.	.
1e−1	0	*28e−1*	*14e−1*	*43e−1*	4.5e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f105 in 5-D**, N=15, mFE=5001					**f105 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	4.2e2	2.0e2	6.4e2	4.2e2	0	*32e+2*	*19e+2*	*95e+2*	4.0e3
1	1	7.0e4	3.3e4	>7e4	5.0e3	.	.	.	.	.
1e−1	1	7.0e4	3.3e4	>7e4	5.0e3	.	.	.	.	.
1e−3	0	*32e−1*	*15e−1*	*46e−1*	4.0e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f106 in 5-D**, N=15, mFE=5001					**f106 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	2.3e2	2.1e2	2.6e2	2.3e2	0	*22e+2*	*76e+1*	*45e+2*	4.5e3
1	6	9.3e3	6.4e3	1.6e4	3.9e3	.	.	.	.	.
1e−1	3	2.0e4	1.2e4	6.1e4	5.0e3	.	.	.	.	.
1e−3	0	*18e−1*	*36e−3*	*46e−1*	3.5e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f107 in 5-D**, N=15, mFE=5001					**f107 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	6.0e1	4.1e1	7.8e1	6.0e1	0	*65e+0*	*53e+0*	*89e+0*	1.8e3
1	12	3.0e3	2.1e3	4.1e3	2.1e3	.	.	.	.	.
1e−1	4	1.6e4	1.0e4	3.4e4	4.6e3	.	.	.	.	.
1e−3	0	*14e−2*	*54e−4*	*21e−1*	3.5e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f108 in 5-D**, N=15, mFE=5001					**f108 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	3.0e2	1.9e2	4.2e2	3.0e2	0	*67e+0*	*55e+0*	*99e+0*	2.2e3
1	2	3.4e4	1.8e4	>7e4	5.0e3	.	.	.	.	.
1e−1	1	7.2e4	3.5e4	>7e4	5.0e3	.	.	.	.	.
1e−3	0	*25e−1*	*67e−2*	*40e−1*	3.2e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f109 in 5-D**, N=15, mFE=5001					**f109 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	2.7e1	2.4e1	2.9e1	2.7e1	8	5.3e3	3.8e3	7.6e3	3.4e3
1	15	5.7e1	4.6e1	6.8e1	5.7e1	0	*94e−1*	*15e−1*	*39e+0*	1.3e3
1e−1	14	7.8e2	2.0e2	1.3e3	7.7e2	.	.	.	.	.
1e−3	3	2.0e4	1.2e4	6.0e4	5.0e3	.	.	.	.	.
1e−5	0	*10e−3*	*11e−5*	*76e−3*	5.6e2	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f110 in 5-D**, N=15, mFE=5001					**f110 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	7	8.0e3	5.3e3	1.3e4	3.1e3	0	*36e+3*	*20e+3*	*49e+3*	2.8e3
1	1	7.2e4	3.4e4	>7e4	5.0e3	.	.	.	.	.
1e−1	0	*15e+0*	*24e−1*	*13e+1*	2.8e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f111 in 5-D**, N=15, mFE=5001					**f111 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	0	*10e+1*	*18e+0*	*27e+1*	2.5e3	0	*34e+3*	*16e+3*	*54e+3*	2.2e3
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f112 in 5-D**, N=15, mFE=5001					**f112 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	2.8e2	2.0e2	3.7e2	2.8e2	0	*15e+2*	*70e+1*	*52e+2*	4.5e3
1	0	*34e−1*	*17e−1*	*43e−1*	2.2e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f113 in 5-D**, N=15, mFE=5001					**f113 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	1.0e3	7.1e2	1.3e3	1.0e3	0	*32e+1*	*20e+1*	*50e+1*	3.5e3
1	2	3.5e4	1.8e4	>7e4	5.0e3	.	.	.	.	.
1e−1	0	*30e−1*	*92e−2*	*53e−1*	2.8e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f114 in 5-D**, N=15, mFE=5001					**f114 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	13	2.7e3	1.9e3	3.6e3	2.3e3	0	*37e+1*	*30e+1*	*44e+1*	2.0e3
1	0	*56e−1*	*20e−1*	*11e+0*	2.2e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f115 in 5-D**, N=15, mFE=5001					**f115 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	2.9e2	1.3e2	4.8e2	2.9e2	0	*14e+1*	*99e+0*	*26e+1*	3.2e3
1	6	8.3e3	5.3e3	1.5e4	2.9e3	.	.	.	.	.
1e−1	0	*11e−1*	*32e−2*	*47e−1*	2.0e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f116 in 5-D**, N=15, mFE=5001					**f116 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	0	*83e+0*	*35e+0*	*21e+1*	3.2e3	0	*18e+3*	*10e+3*	*23e+3*	3.5e3
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f117 in 5-D**, N=15, mFE=5001					**f117 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	0	*21e+1*	*99e+0*	*49e+1*	3.2e3	0	*24e+3*	*16e+3*	*36e+3*	3.2e3
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f118 in 5-D**, N=15, mFE=5001					**f118 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	2	3.7e4	1.9e4	>7e4	5.0e3	0	*54e+2*	*27e+2*	*73e+2*	5.0e3
1	0	*33e+0*	*79e−1*	*88e+0*	3.5e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f119 in 5-D**, N=15, mFE=5001					**f119 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	2.0e1	1.4e1	2.6e1	2.0e1	0	*18e+0*	*15e+0*	*23e+0*	3.5e3
1	12	2.7e3	1.9e3	3.6e3	2.2e3	.	.	.	.	.
1e−1	3	2.4e4	1.4e4	7.1e4	5.0e3	.	.	.	.	.
1e−3	0	*51e−2*	*48e−3*	*14e−1*	3.2e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f120 in 5-D**, N=15, mFE=5001					**f120 in 20-D**, N=15, mFE=5066				
	#	ERT	10%	90%	RT$_{succ}$	#	ERT	10%	90%	RT$_{succ}$
10	15	1.9e1	1.3e1	2.6e1	1.9e1	0	*19e+0*	*13e+0*	*23e+0*	2.5e3
1	6	9.6e3	7.2e3	1.5e4	5.0e3	.	.	.	.	.
1e−1	0	*13e−1*	*23e−2*	*23e−1*	3.2e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
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 $f_{\rm opt} + \Delta f$ (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$_{\rm succ}$). If $f_{\rm opt} + \Delta f$ 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.

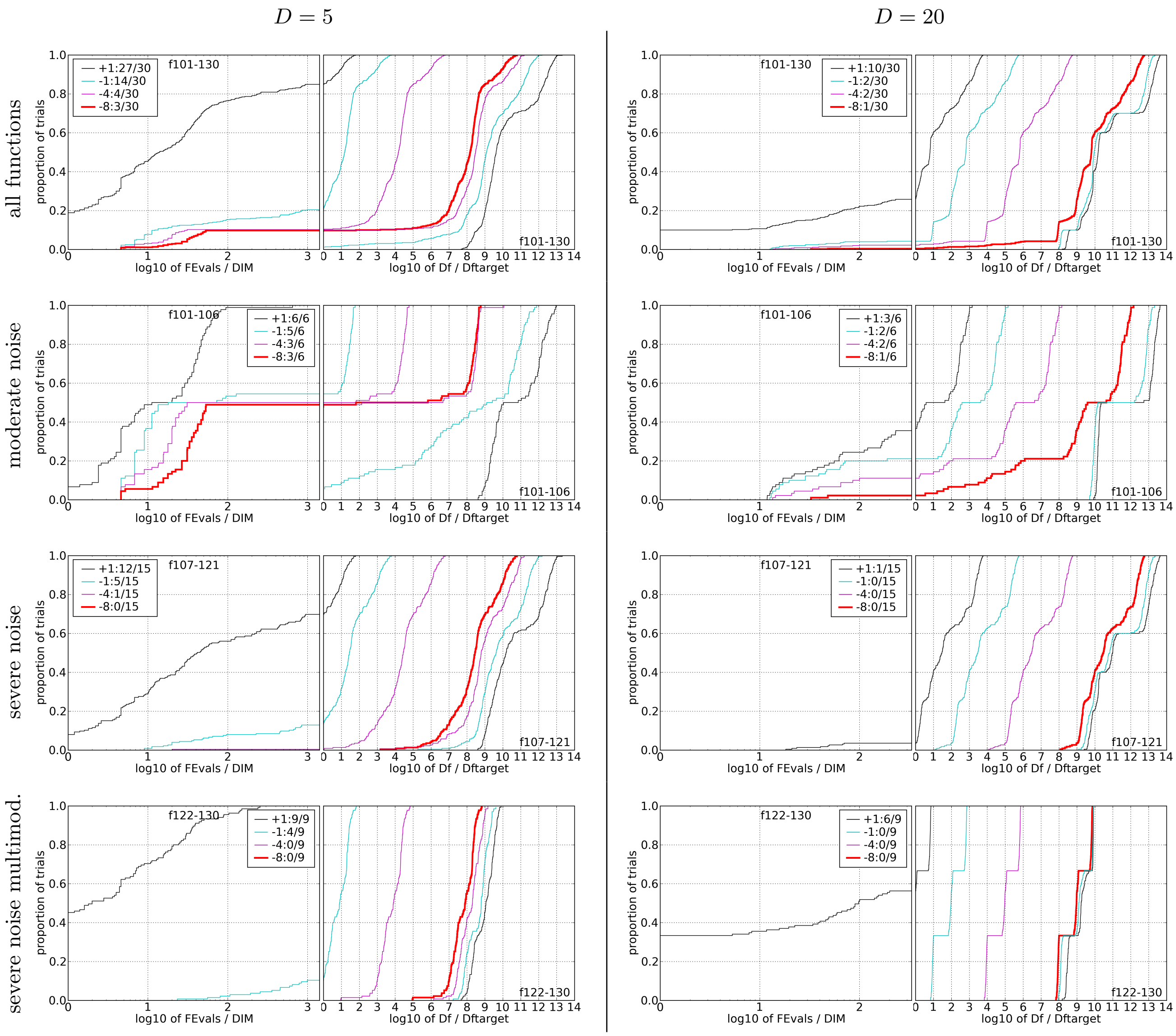

Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left) or Δf. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\rm opt} + \Delta f$ with $\Delta f = 10^k$, 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: 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=5001					**f_{121} in 20-D**, N=15, mFE=5066				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.4e1	9.5e0	1.8e1	1.4e1	0	*16e+0*	*13e+0*	*22e+0*	3.5e3
1	15	2.1e2	1.3e2	3.0e2	2.1e2	.	.	.	.	.
1e−1	7	6.7e3	4.7e3	1.0e4	3.6e3	.	.	.	.	.
1e−3	0	*11e−2*	*23e−3*	*59e−2*	2.5e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{122} in 5-D, N=15, mFE=5001					**f_{122} in 20-D**, N=15, mFE=5066				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.0e1	1.5e1	2.4e1	2.0e1	9	5.0e3	3.7e3	7.1e3	3.3e3
1	3	2.2e4	1.3e4	6.4e4	5.0e3	0	*98e−1*	*88e−1*	*12e+0*	1.8e3
1e−1	0	*20e−1*	*62e−2*	*27e−1*	2.5e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{123} in 5-D, N=15, mFE=5001					**f_{123} in 20-D**, N=15, mFE=5066				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.9e1	2.3e1	5.5e1	3.9e1	7	7.5e3	5.0e3	1.2e4	3.3e3
1	0	*25e−1*	*15e−1*	*32e−1*	2.8e3	0	*10e+0*	*79e−1*	*12e+0*	1.8e3
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{124} in 5-D, N=15, mFE=5001					**f_{124} in 20-D**, N=15, mFE=5066				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.0e1	7.8e0	1.3e1	1.0e1	15	1.3e3	1.0e3	1.6e3	1.3e3
1	10	3.7e3	2.7e3	5.3e3	2.6e3	0	*85e−1*	*70e−1*	*97e−1*	1.8e3
1e−1	0	*88e−2*	*38e−2*	*15e−1*	2.8e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{125} in 5-D, N=15, mFE=5001					**f_{125} in 20-D**, N=15, mFE=5066				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.3e0	1.1e0	1.4e0	1.3e0	15	1.1e0	1.0e0	1.1e0	1.1e0
1	15	3.0e1	2.2e1	3.9e1	3.0e1	14	1.6e3	1.1e3	2.2e3	1.5e3
1e−1	5	1.1e4	6.8e3	2.2e4	3.1e3	0	*91e−2*	*74e−2*	*99e−2*	1.8e3
1e−3	0	*13e−2*	*54e−3*	*26e−2*	2.5e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{126} in 5-D, N=15, mFE=5001					**f_{126} in 20-D**, N=15, mFE=5066				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.3e0	1.1e0	1.4e0	1.3e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	15	9.0e1	5.7e1	1.3e2	9.0e1	14	1.4e3	9.2e2	2.0e3	1.4e3
1e−1	0	*19e−2*	*13e−2*	*30e−2*	2.5e3	0	*86e−2*	*75e−2*	*95e−2*	2.2e3
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{127} in 5-D, N=15, mFE=5001					**f_{127} in 20-D**, N=15, mFE=5066				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.3e0	1.1e0	1.5e0	1.3e0	15	1.1e0	1.0e0	1.3e0	1.1e0
1	15	2.4e1	1.9e1	3.0e1	2.4e1	14	1.7e3	1.2e3	2.2e3	1.7e3
1e−1	3	2.2e4	1.3e4	6.9e4	3.9e3	0	*87e−2*	*77e−2*	*93e−2*	2.2e3
1e−3	0	*20e−2*	*94e−3*	*33e−2*	2.2e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{128} in 5-D, N=15, mFE=5001					**f_{128} in 20-D**, N=15, mFE=5066				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.7e2	1.7e2	3.8e2	2.7e2	0	*68e+0*	*62e+0*	*71e+0*	3.5e3
1	5	1.4e4	8.8e3	2.5e4	4.0e3	.	.	.	.	.
1e−1	2	3.7e4	1.8e4	>7e4	4.1e3	.	.	.	.	.
1e−3	1	7.5e4	3.7e4	>7e4	4.8e3	.	.	.	.	.
1e−5	0	*18e−1*	*39e−3*	*42e−1*	3.5e3	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{129} in 5-D, N=15, mFE=5001					**f_{129} in 20-D**, N=15, mFE=5066				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.4e2	2.3e2	4.4e2	3.4e2	0	*67e+0*	*55e+0*	*70e+0*	3.5e3
1	0	*29e−1*	*19e−1*	*60e−1*	2.5e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{130} in 5-D, N=15, mFE=5001					**f_{130} in 20-D**, N=15, mFE=5066				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	8.2e1	6.7e1	9.8e1	8.2e1	0	*62e+0*	*55e+0*	*69e+0*	8.9e2
1	11	3.1e3	2.2e3	3.9e3	2.9e3	.	.	.	.	.
1e−1	4	1.6e4	1.1e4	3.3e4	5.0e3	.	.	.	.	.
1e−3	1	7.5e4	3.7e4	>7e4	5.0e3	.	.	.	.	.
1e−5	0	*39e−2*	*14e−3*	*20e−1*	2.8e3	.	.	.	.	.
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.