BBO-Benchmarking of the GLOBAL method for the Noiseless Function Testbed

László Pál
Sapientia - Hungarian University of Transylvania
530140 Miercurea-Ciuc, Piata Libertatii, Nr. 1, Romania
pallaszlo@sapientia.siculorum.ro

Tibor Csendes
University of Szeged
6701 Szeged, P.O. Box 652, Hungary
csendes@inf.u-szeged.hu

Mihály Csaba Markót
University of Wien, Faculty of Mathematics
Nordbergstraße 15
1090 Wien, Austria
Mihaly.Markot@univie.ac.at

Arnold Neumaier
University of Wien, Faculty of Mathematics
Nordbergstraße 15
1090 Wien, Austria
Arnold.Neumaier@univie.ac.at

ABSTRACT

GLOBAL is a multistart type stochastic method for bound constrained global optimization problems. Its goal is to find all the local minima that are potentially global. For this reason it involves a combination of sampling, clustering, and local search. We report its results on the noisy free problems given.

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

Keywords

Benchmarking, Black-box optimization, Clustering

1. INTRODUCTION

The multistart clustering global optimization method called GLOBAL [2] has been introduced in the 80s for bound constrained global optimization problems with black-box type objective functions. The algorithm is based on Boender's algorithm [1] and its goal is to find all local minimizer points that are potentially global. The local search procedure used by GLOBAL was originally either a quasi-Newton procedure with the DFP update formula or a random walk type direct search method called UNIRANDI [7]. GLOBAL was originally coded in Fortran and C languages.

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

Based on the old GLOBAL method we introduced a new version [3] coded in MATLAB. The algorithm was carefully studied and it was modified in some places to achieve better reliability and efficiency while allowing higher dimensional problems to be solved. In the new version we use the quasi-Newton local search method with the BFGS update instead of the earlier DFP. We also combined GLOBAL with other local search methods like the Nelder-Mead simplex method. All three versions (Fortran, C, MATLAB) of the algorithm are freely available for academic and nonprofit purposes at `www.inf.u-szeged.hu/∼csendes/regist.php` (after registration and limited for low dimensional problems).

In this paper, the algorithm is benchmarked on the noisyfree BBOB 2009 testbed [4, 6] according to the experimental design from [5].

2. ALGORITHM PRESENTATION

The GLOBAL method has two phases: a global and a local one. The global phase consists of sampling and clustering, while the local phase is based on local searches. The local minimizer points are found by means of a local search procedure, starting from appropriately chosen points from the sample drawn uniformly within the set of feasibility. In an effort to identify the region of attraction of a local minimizer, the procedure invokes a clustering procedure. The main steps of GLOBAL are summarized in Algorithm 1.

3. EXPERIMENTAL PROCEDURE

GLOBAL has six parameters to set: the number of sample points, the number of best points selected, the stopping criterion parameter for the local search, the maximum number of function evaluations for local search, the maximum number of local minima to explore, and the used local method. All these parameters have a default value and usually it is enough to change only the first three of them.

In all dimensions and functions we used 300 sample points, and the 2 best points. In 2, 3 and 5 dimensions the local search tolerance was 10^{-8}, the maximum number of function evaluations for local search was 5000 and the local search was the simplex method. In 10 and 20 dimensions with the

Algorithm 1 A concise description of the GLOBAL optimization algorithm

Step 1: Draw N points with uniform distribution in X, and add them to the current cumulative sample C. Construct the transformed sample T by taking the γ percent of the points in C with the lowest function value.

Step 2: Apply the clustering procedure to T one by one. If all points of T can be assigned to an existing cluster, go to Step 4.

Step 3: Apply the local search procedure to the points in T not yet clustered. Repeat Step 3 until every point has been assigned to a cluster.

Step 4: If a new local minimizer has been found, go to Step 1.

Step 5: Determine the smallest local minimum value found, and stop.

3,4,7,16,23 functions we used the previous settings with local search tolerance 10^{-9}. In the case of the remained functions we used the previous parameters with 10000 number of maximum function evaluation and with the BFGS local search method.

The corresponding crafting effort is: $CrE_{10} = CrE_{20} = -(\frac{5}{24} \ln \frac{5}{24} + \frac{19}{24} \ln \frac{19}{24}) = 0.5117$.

4. CPU TIMING EXPERIMENT

For the timing experiment the GLOBAL algorithm was run on f8 and restarted until at least 30 seconds had passed (according to Figure 2 in [5]). These experiments have been conducted with an Intel Core 2 Duo 2.00 GHz under Windows XP using MATLAB 7.6.0.324 version. We have done two experiments using the BFGS and the simplex local search methods. The other algorithm parameters were the same. In the first case (BFGS) the results were $(2.8, 2.9, 3.0, 3.0, 3.2, 3.2) \cdot 10^{-4}$ seconds, while in the second case (simplex) they were $(2.6, 2.9, 3.4, 4.6, 7.5, 21.0) \cdot 10^{-4}$ seconds per function evaluation in dimensions 2, 3, 5, 10, 20, and 40, respectively.

5. RESULTS

Results from experiments according to [5] on the benchmark functions given in [4, 6] are presented in Figures 1 and 2 and in Table 1.

6. CONCLUSION

We have summarized the results of the GLOBAL stochastic multistart algorithm on the noiseless function testbed. Based on these results we can conclude that GLOBAL performs well on most functions, except those problems which have a high number of local minimizers.

7. REFERENCES

[1] C. G. E. Boender, A. H. G. Rinnooy Kan, G. T. Timmer, and L. Stougie. A stochastic method for global optimization. *Math. Program.*, 22:125–140, 1988.

[2] T. Csendes. Nonlinear parameter estimation by global optimization – efficiency and reliability. *Acta Cybernetica*, 8:361–370, 1988.

[3] T. Csendes, L. Pál, J. O. H. Sendin, and J. R. Banga. The GLOBAL Optimization Method Revisited. *Optimization Letters*, 2:445–454, 2008.

[4] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009.

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

[6] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009.

[7] T. Järvi. A random search optimizer with an application to a max-min problem. *Publications of the Institute for Applied Mathematics*, 3, 1973.

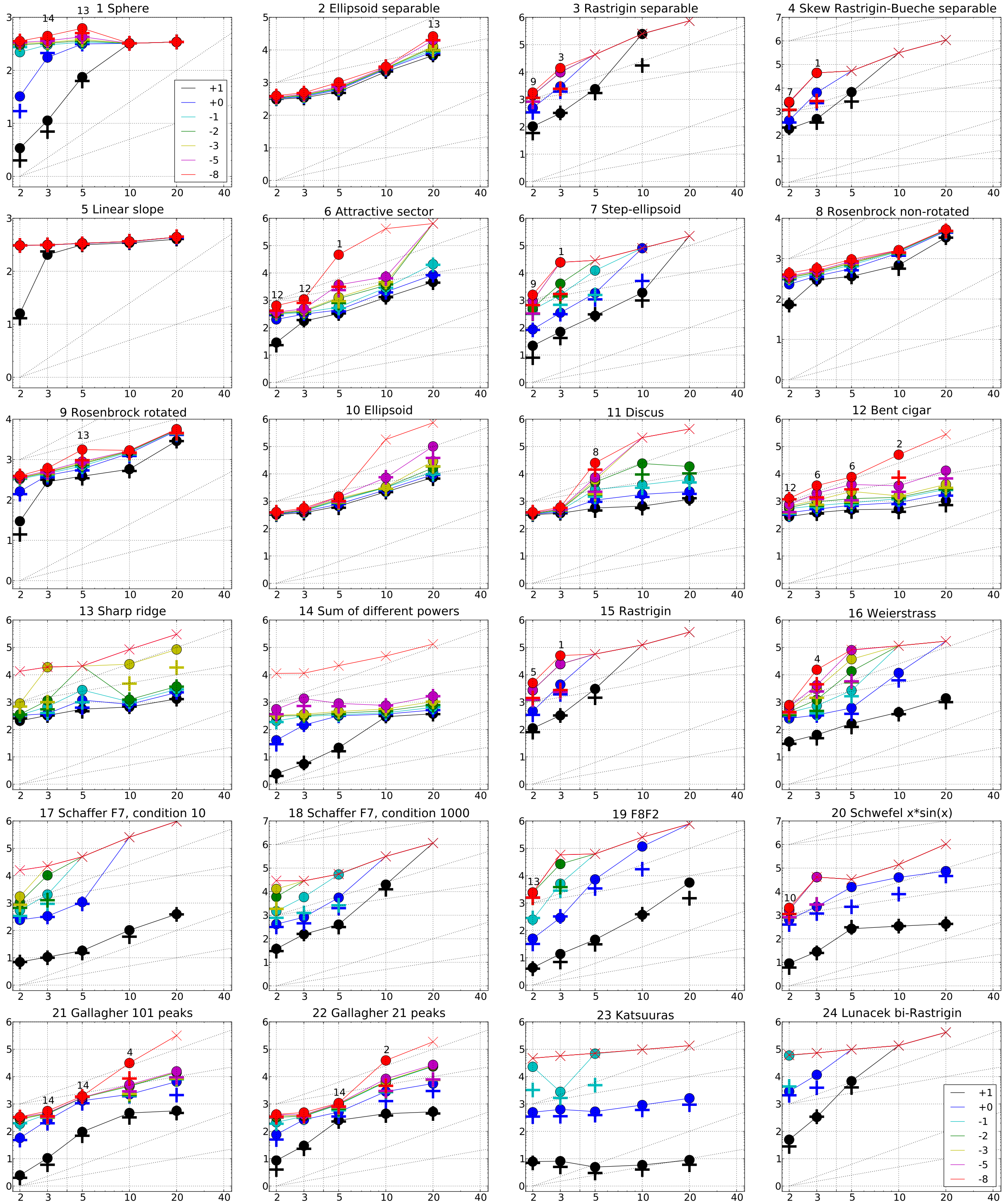

Figure 1: Expected Running Time (ERT, ●) **to reach** $f_{\rm opt} + \Delta f$ **and median number of function evaluations of successful trials** (+), **shown for** $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ **(the exponent is given in the legend of** f_1 **and** f_{24}**) versus 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=793 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_1 in 20-D, N=15, mFE=510 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	7.5e1	5.5e1	9.0e1	7.5e1	15	3.4e2	3.4e2	3.4e2	3.4e2
1	15	3.1e2	3.1e2	3.2e2	3.1e2	15	3.4e2	3.4e2	3.4e2	3.4e2
1e−1	15	3.4e2	3.4e2	3.4e2	3.4e2	15	3.4e2	3.4e2	3.4e2	3.4e2
1e−3	15	3.8e2	3.8e2	3.9e2	3.8e2	15	3.4e2	3.4e2	3.4e2	3.4e2
1e−5	15	4.3e2	4.3e2	4.4e2	4.3e2	15	3.4e2	3.4e2	3.4e2	3.4e2
1e−8	13	6.2e2	5.8e2	6.6e2	5.4e2	15	3.4e2	3.4e2	3.4e2	3.4e2

Δf	f_2 in 5-D, N=15, mFE=1700 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_2 in 20-D, N=15, mFE=50985 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	5.2e2	4.8e2	5.5e2	5.2e2	15	6.8e3	6.2e3	7.0e3	6.8e3
1	15	6.0e2	5.7e2	6.4e2	6.0e2	15	8.7e3	8.2e3	9.0e3	8.7e3
1e−1	15	6.4e2	6.3e2	6.7e2	6.4e2	15	1.0e4	8.8e3	1.1e4	1.0e4
1e−3	15	7.0e2	6.8e2	7.5e2	7.0e2	15	1.3e4	1.1e4	1.4e4	1.3e4
1e−5	15	7.5e2	7.2e2	8.2e2	7.5e2	14	2.0e4	1.5e4	2.5e4	1.9e4
1e−8	15	1.0e3	9.3e2	1.1e3	1.0e3	13	2.6e4	2.4e4	3.2e4	2.3e4

Δf	f_3 in 5-D, N=15, mFE=5805 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_3 in 20-D, N=15, mFE=58220 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	11	2.4e3	1.9e3	3.0e3	2.0e3	0	*15e+1*	*89e+0*	*19e+1*	3.2e4
1	0	*50e−1*	*30e−1*	*13e+0*	8.9e2	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_4 in 5-D, N=15, mFE=7664 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_4 in 20-D, N=15, mFE=89336 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	6	6.7e3	4.3e3	8.4e3	3.1e3	0	*20e+1*	*17e+1*	*30e+1*	4.0e4
1	0	*11e+0*	*60e−1*	*19e+0*	8.9e2	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_5 in 5-D, N=15, mFE=381 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_5 in 20-D, N=15, mFE=510 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.2e2	3.1e2	3.2e2	3.2e2	15	4.0e2	4.0e2	4.1e2	4.0e2
1	15	3.3e2	3.3e2	3.4e2	3.3e2	15	4.4e2	4.3e2	4.4e2	4.4e2
1e−1	15	3.4e2	3.3e2	3.5e2	3.4e2	15	4.4e2	4.4e2	4.5e2	4.4e2
1e−3	15	3.4e2	3.3e2	3.4e2	3.4e2	15	4.4e2	4.3e2	4.5e2	4.4e2
1e−5	15	3.4e2	3.3e2	3.5e2	3.4e2	15	4.4e2	4.3e2	4.5e2	4.4e2
1e−8	15	3.4e2	3.3e2	3.4e2	3.4e2	15	4.4e2	4.3e2	4.6e2	4.4e2

Δf	f_6 in 5-D, N=15, mFE=4310 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_6 in 20-D, N=15, mFE=50685 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.3e2	3.0e2	3.4e2	3.3e2	15	4.6e3	4.4e3	5.2e3	4.6e3
1	15	4.4e2	4.2e2	4.8e2	4.4e2	15	8.5e3	7.9e3	9.5e3	8.5e3
1e−1	15	5.6e2	5.1e2	6.0e2	5.6e2	15	2.1e4	1.8e4	2.5e4	2.1e4
1e−3	13	1.3e3	1.1e3	1.6e3	1.2e3	0	*42e−3*	*22e−3*	*80e−3*	2.8e4
1e−5	9	3.7e3	3.1e3	4.2e3	2.5e3	.	.	.	.	.
1e−8	1	4.7e4	4.1e4	4.9e4	3.8e3	.	.	.	.	.

Δf	f_7 in 5-D, N=15, mFE=3471 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_7 in 20-D, N=15, mFE=24177 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.7e2	2.2e2	3.2e2	2.7e2	0	*22e+0*	*13e+0*	*39e+0*	6.3e3
1	9	1.8e3	1.6e3	2.0e3	1.0e3	.	.	.	.	.
1e−1	2	1.2e4	1.0e4	1.4e4	1.4e3	.	.	.	.	.
1e−3	0	*82e−2*	*33e−3*	*21e−1*	1.0e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_8 in 5-D, N=15, mFE=1687 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_8 in 20-D, N=15, mFE=10611 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.6e2	3.6e2	3.8e2	3.6e2	15	3.3e3	3.1e3	3.6e3	3.3e3
1	15	5.7e2	4.5e2	6.6e2	5.7e2	15	4.8e3	4.4e3	4.9e3	4.8e3
1e−1	15	7.0e2	6.5e2	8.2e2	7.0e2	15	5.0e3	4.6e3	5.2e3	5.0e3
1e−3	15	8.2e2	7.0e2	8.9e2	8.2e2	15	5.2e3	4.6e3	5.6e3	5.2e3
1e−5	15	8.7e2	7.8e2	9.0e2	8.7e2	15	5.2e3	4.7e3	6.1e3	5.2e3
1e−8	15	9.6e2	8.7e2	1.0e3	9.6e2	15	5.3e3	4.7e3	5.9e3	5.3e3

Δf	f_9 in 5-D, N=15, mFE=4356 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_9 in 20-D, N=15, mFE=12927 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.9e2	3.5e2	4.4e2	3.9e2	15	2.9e3	2.8e3	3.0e3	2.9e3
1	15	5.8e2	5.4e2	6.7e2	5.8e2	15	5.2e3	4.6e3	5.8e3	5.2e3
1e−1	15	6.9e2	6.6e2	7.3e2	6.9e2	15	5.4e3	4.7e3	5.8e3	5.4e3
1e−3	15	8.3e2	8.0e2	9.6e2	8.3e2	15	5.5e3	4.6e3	6.3e3	5.5e3
1e−5	15	8.9e2	8.4e2	9.4e2	8.9e2	15	5.6e3	4.4e3	6.1e3	5.6e3
1e−8	13	1.8e3	1.6e3	2.1e3	1.6e3	15	5.6e3	5.0e3	6.8e3	5.6e3

Δf	f_{10} in 5-D, N=15, mFE=2979 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{10} in 20-D, N=15, mFE=91227 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	6.6e2	5.9e2	7.1e2	6.6e2	15	7.4e3	7.0e3	7.6e3	7.4e3
1	15	7.9e2	7.4e2	9.0e2	7.9e2	15	9.4e3	8.5e3	1.0e4	9.4e3
1e−1	15	1.0e3	8.4e2	1.1e3	1.0e3	15	1.2e4	1.1e4	1.3e4	1.2e4
1e−3	15	1.3e3	9.9e2	1.5e3	1.3e3	12	2.9e4	2.2e4	3.4e4	2.1e4
1e−5	15	1.4e3	1.2e3	1.6e3	1.4e3	6	1.0e5	8.3e4	1.1e5	3.3e4
1e−8	15	1.5e3	1.3e3	1.7e3	1.5e3	0	*19e−6*	*15e−7*	*60e−4*	2.5e4

Δf	f_{11} in 5-D, N=15, mFE=25894 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{11} in 20-D, N=15, mFE=74094 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	5.6e2	5.0e2	6.6e2	5.6e2	15	1.2e3	1.1e3	1.2e3	1.2e3
1	15	1.1e3	8.7e2	1.2e3	1.1e3	15	2.3e3	1.8e3	2.5e3	2.3e3
1e−1	15	2.7e3	1.7e3	3.5e3	2.7e3	15	6.3e3	4.1e3	7.6e3	6.3e3
1e−3	13	5.9e3	4.1e3	8.6e3	4.3e3	0	*74e−4*	*29e−4*	*21e−3*	1.0e4
1e−5	13	7.4e3	4.7e3	8.9e3	5.2e3	.	.	.	.	.
1e−8	8	2.5e4	2.1e4	3.0e4	1.2e4	.	.	.	.	.

Δf	f_{12} in 5-D, N=15, mFE=6299 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{12} in 20-D, N=15, mFE=23553 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	4.9e2	4.5e2	5.3e2	4.9e2	15	1.0e3	8.2e2	1.3e3	1.0e3
1	15	7.1e2	5.3e2	8.7e2	7.1e2	15	1.9e3	1.5e3	2.2e3	1.9e3
1e−1	15	8.8e2	7.5e2	9.8e2	8.8e2	15	2.7e3	2.4e3	3.2e3	2.7e3
1e−3	12	2.3e3	1.6e3	3.0e3	1.8e3	15	4.1e3	2.9e3	5.8e3	4.1e3
1e−5	9	4.1e3	2.8e3	5.1e3	2.8e3	11	1.3e4	1.0e4	1.6e4	8.9e3
1e−8	6	7.7e3	6.0e3	9.4e3	2.6e3	0	*67e−8*	*79e−9*	*20e−5*	1.0e4

Δf	f_{13} in 5-D, N=15, mFE=3073 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{13} in 20-D, N=15, mFE=24204 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	5.5e2	4.9e2	6.7e2	5.5e2	15	1.3e3	1.3e3	1.4e3	1.3e3
1	11	1.2e3	1.1e3	1.5e3	9.1e2	15	2.2e3	2.2e3	2.3e3	2.2e3
1e−1	6	2.8e3	2.5e3	3.1e3	1.0e3	15	2.9e3	2.9e3	2.9e3	2.9e3
1e−3	0	*19e−2*	*18e−3*	*38e−1*	6.3e2	3	8.4e4	7.1e4	9.2e4	1.4e4
1e−5	.	.	.	.	.	0	*16e−4*	*45e−6*	*33e−4*	8.9e3
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{14} in 5-D, N=15, mFE=2078 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{14} in 20-D, N=15, mFE=10701 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.1e1	1.9e1	2.9e1	2.1e1	15	3.8e2	3.7e2	3.8e2	3.8e2
1	15	3.2e2	3.2e2	3.3e2	3.2e2	15	5.2e2	5.0e2	5.3e2	5.2e2
1e−1	15	3.5e2	3.4e2	3.5e2	3.5e2	15	6.4e2	6.2e2	6.7e2	6.4e2
1e−3	15	4.6e2	4.4e2	4.8e2	4.6e2	15	1.0e3	1.0e3	1.1e3	1.0e3
1e−5	13	9.0e2	8.1e2	1.0e3	8.2e2	15	1.6e3	1.6e3	1.7e3	1.6e3
1e−8	0	*59e−7*	*44e−7*	*13e−6*	7.1e2	0	*28e−7*	*18e−7*	*44e−7*	6.3e3

Δf	f_{15} in 5-D, N=15, mFE=10869 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{15} in 20-D, N=15, mFE=26514 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	11	3.0e3	2.0e3	4.2e3	2.5e3	0	*25e+1*	*19e+1*	*28e+1*	1.8e4
1	0	*90e−1*	*40e−1*	*14e+0*	1.4e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{16} in 5-D, N=15, mFE=6135 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{16} in 20-D, N=15, mFE=12339 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.7e2	1.2e2	1.9e2	1.7e2	15	1.4e3	9.3e2	1.8e3	1.4e3
1	15	6.1e2	4.8e2	7.5e2	6.1e2	0	*42e−1*	*28e−1*	*60e−1*	6.3e3
1e−1	12	2.7e3	1.9e3	3.4e3	2.3e3	.	.	.	.	.
1e−3	2	3.7e4	3.5e4	4.2e4	5.9e3	.	.	.	.	.
1e−5	1	8.0e4	7.1e4	8.5e4	5.8e3	.	.	.	.	.
1e−8	0	*18e−3*	*20e−5*	*31e−2*	2.2e3	.	.	.	.	.

Δf	f_{17} in 5-D, N=15, mFE=6949 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{17} in 20-D, N=15, mFE=78903 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.8e1	1.4e1	2.2e1	1.8e1	15	3.9e2	3.8e2	4.2e2	3.9e2
1	13	1.1e3	9.5e2	1.3e3	9.7e2	0	*44e−1*	*30e−1*	*55e−1*	2.8e4
1e−1	0	*47e−2*	*16e−2*	*11e−1*	1.0e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{18} in 5-D, N=15, mFE=8792 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{18} in 20-D, N=15, mFE=97425 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	4.0e2	3.6e2	5.3e2	4.0e2	0	*17e+0*	*15e+0*	*23e+0*	4.5e4
1	7	5.5e3	4.2e3	6.0e3	3.5e3	.	.	.	.	.
1e−1	1	5.4e4	3.6e4	6.7e4	4.5e3	.	.	.	.	.
1e−3	0	*12e−1*	*46e−2*	*32e−1*	1.8e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{19} in 5-D, N=15, mFE=5482 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{19} in 20-D, N=15, mFE=60465 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	4.6e1	2.5e1	5.6e1	4.6e1	14	5.6e3	2.5e3	8.5e3	5.5e3
1	6	7.3e3	6.0e3	8.8e3	2.5e3	0	*57e−1*	*40e−1*	*84e−1*	3.2e4
1e−1	0	*10e−1*	*22e−2*	*17e−1*	1.8e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{20} in 5-D, N=15, mFE=3836 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{20} in 20-D, N=15, mFE=84276 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.7e2	2.4e2	2.9e2	2.7e2	15	4.3e2	4.2e2	4.3e2	4.3e2
1	2	1.6e4	1.5e4	1.8e4	2.4e3	9	7.5e4	5.8e4	9.0e4	4.3e4
1e−1	0	*13e−1*	*95e−2*	*19e−1*	8.9e2	0	*99e−2*	*82e−2*	*12e−1*	2.8e4
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{21} in 5-D, N=15, mFE=4402 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{21} in 20-D, N=15, mFE=45129 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	9.6e1	7.2e1	1.1e2	9.6e1	15	5.6e2	4.9e2	6.5e2	5.6e2
1	15	1.3e3	1.2e3	1.6e3	1.3e3	13	6.5e3	4.0e3	7.0e3	5.6e3
1e−1	14	1.7e3	1.4e3	2.0e3	1.6e3	10	1.4e4	1.1e4	1.7e4	1.0e4
1e−3	14	1.7e3	1.3e3	1.9e3	1.6e3	10	1.5e4	1.1e4	1.6e4	1.1e4
1e−5	14	1.7e3	1.4e3	1.9e3	1.7e3	10	1.6e4	1.6e4	1.9e4	1.1e4
1e−8	14	1.9e3	1.5e3	2.3e3	1.9e3	0	*25e−8*	*12e−9*	*18e−1*	1.0e4

Δf	f_{22} in 5-D, N=15, mFE=2700 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{22} in 20-D, N=15, mFE=31137 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.5e2	1.9e2	3.4e2	2.5e2	15	5.2e2	4.9e2	5.9e2	5.2e2
1	15	5.1e2	4.2e2	5.4e2	5.1e2	12	5.6e3	3.7e3	8.9e3	4.0e3
1e−1	14	9.4e2	7.7e2	1.1e3	9.1e2	5	2.3e4	1.9e4	2.6e4	8.1e3
1e−3	14	1.0e3	9.0e2	1.3e3	9.7e2	5	2.5e4	1.9e4	3.0e4	8.9e3
1e−5	14	1.0e3	7.9e2	1.3e3	1.0e3	5	2.7e4	2.3e4	3.2e4	9.8e3
1e−8	14	1.1e3	8.5e2	1.3e3	1.0e3	0	*69e−2*	*50e−8*	*51e−1*	3.2e3

Δf	f_{23} in 5-D, N=15, mFE=5298 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{23} in 20-D, N=15, mFE=10437 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	4.9e0	3.7e0	6.6e0	4.9e0	15	8.9e0	7.5e0	1.1e1	8.9e0
1	15	5.3e2	4.6e2	6.4e2	5.3e2	15	1.6e3	9.7e2	1.8e3	1.6e3
1e−1	1	6.9e4	6.4e4	7.4e4	4.3e3	0	*43e−2*	*24e−2*	*65e−2*	4.0e3
1e−3	0	*23e−2*	*14e−2*	*35e−2*	2.2e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{24} in 5-D, N=15, mFE=10787 #	ERT	10%	90%	$\mathrm{RT_{succ}}$	f_{24} in 20-D, N=15, mFE=29466 #	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	9	6.8e3	5.3e3	8.6e3	4.5e3	0	*21e+1*	*19e+1*	*23e+1*	1.1e4
1	0	*91e−1*	*52e−1*	*12e+0*	2.0e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
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 $f_{\mathrm{opt}}+\Delta f$ (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.

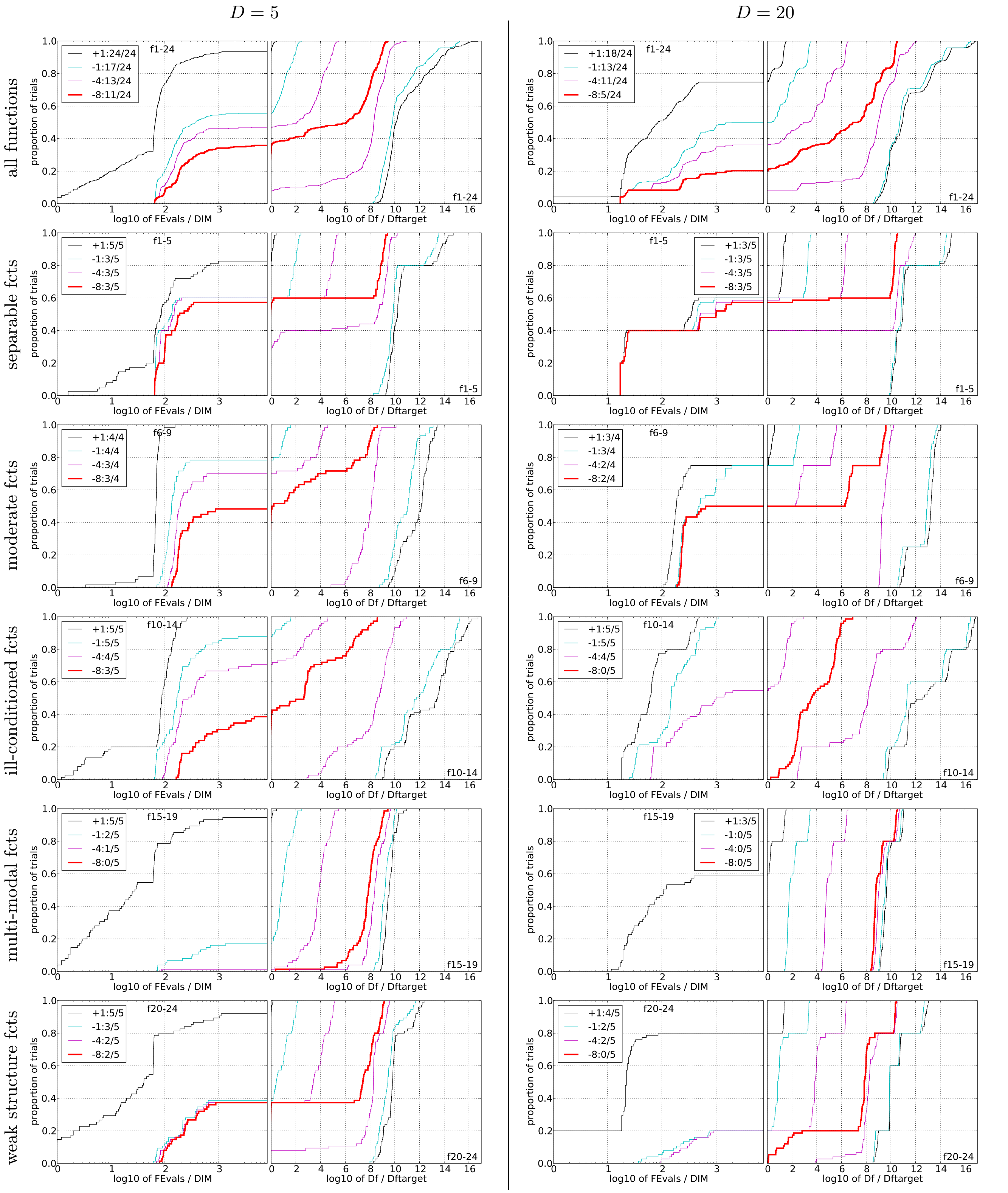

Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left) or Δf. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\rm opt} + \Delta f$ with $\Delta f = 10^k$,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