BBOB-Benchmarking Two Variants of the Line-Search Algorithm

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 line search, or coordinate-wise search, algorithm is tested on the BBOB 2009 testbed. Two different univariate search algorithms (`fminbnd` from MATLAB and STEP) were tried and compared. The results are as expected: line search method can optimize only separable functions, for other functions it fails. The STEP method is slightly slower, however, is more robust in the multimodal case. The line search algorithms also identified 2 functions of the test suite (in addition to separable problems) which might be easy for algorithms exploiting separability.

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, Evolutionary computation, Brent's method, fminbnd, STEP

1. INTRODUCTION

The line search algorithm is one of the basic and simplest optimization algorithms. In any comparison, it should serve as a baseline algorithm. It is pretty efficient for separable functions. The results of the line search algorithm should thus discover test functions which are separable or functions which are easy for algorithms exploiting separability.

It is not expected that the line search algorithm will be effective on non-separable functions. This should demonstrate the need to construct algorithms which can solve non-separable functions.

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

2. ALGORITHM DESCRIPTION

The line search heuristic is pretty simple. It starts from a randomly selected point. Then it iterates through individual directions and optimizes the function with respect to the chosen direction, keeping the other solution components fixed. After finding the optimum in one direction, it moves to the best solution found and switches to another direction. If the solution does not change after going through all the directions, a local optimum is found and the algorithm finishes.

In this paper, the multistart version of line search is considered; each launch begins in a different randomly selected initial point.

The efficiency of the line search greatly depends on the procedure employed for the univariate bounded optimization. In this paper, two univariate optimization algorithms are compared: the MATLAB function `fminbnd`, and the STEP algorithm.

2.1 MATLAB `fminbnd` function

The MATLAB `fminbnd` function (revision 1.18.4.11 was used) is based on golden-section search and parabolic interpolation. It is able to identify the optimum of quadratic functions in a few steps. On the other hand, it is a local search technique, it can miss the global optimum (of the 1D function). Since this is a rather standard ready-to-use algorithm, it will not be described here in more detail.

2.2 STEP

The acronym STEP stands for *select the easiest point.* The STEP method [4] is univariate global search algorithm based on interval division. It starts from one interval initialized with x_l and x_u, lower and upper bound of the interval, with both points evaluated. In each iteration it selects one interval and divides it to halves by sampling and evaluating the point in the middle.

The STEP algorithm selects the next interval to sample from by evaluating its difficulty, i.e. by its belief how difficult it would be to improve the best-so-far solution by sampling from the respective interval. The measure of interval difficulty chosen in STEP is the value of the coefficient a from a quadratic function $f(x) = ax^2 + bx + c$ which goes through both boundary points and somewhere on the interval reaches the value of $f_{\mathrm{BSF}} - \epsilon$ (ϵ is a small positive number, typically from 10^{-3} to 10^{-8}, meaning an improvement of f_{BSF} by a nontrivial amount). Example of a few STEP iterations can be seen in Fig. 1.

Note that this method is also based on parabolic inter-

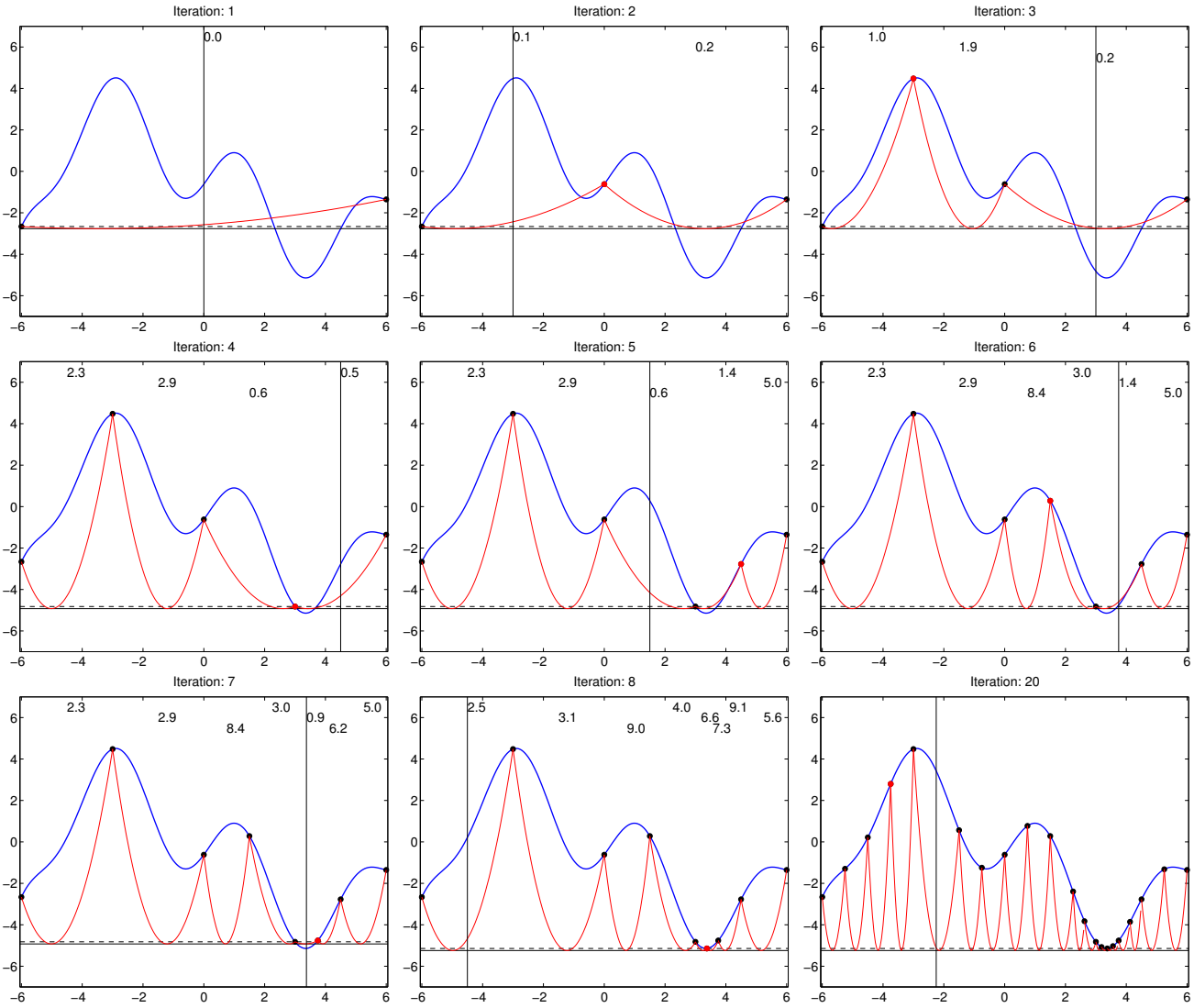

Figure 1: Demonstration of the behavior of the STEP algorithm. Blue line: the objective function. Black dots: previously sampled data points, interval boundaries. Dashed black horizontal line: f_{BSF} level. Solid black horizontal line: $f_{\mathrm{BSF}} - \epsilon$ level. Red line: parabolic functions going thhrough the interval boundaries and touching the objective level $f_{\mathrm{BSF}} - \epsilon$. Red dot: last sampled data point. Vertical line: the place where next point will be sampled. Numbers: the difficulty indices of the respective intervals—the coefficients a of the respective parabolas.

polation (similarly to `fminbnd`), however, applied in a completely different manner than in `fminbnd`. In `fminbnd` it is assumed, that the objective function really has a parabolic shape and the parabola interpolates 3 points on the function. STEP does not assume that the objective function is quadratic; parabola was chosen as the measure of interval difficulty since it is the smoothest function (its curvature does not change, it is a constant value $2a$). It interpolates only 2 points on the function and is required to reach the value $f_{\mathrm{BSF}} - \epsilon$ between the two respective points.

2.3 Parameter Settings

The `fminbnd` function does not have any parameters (except the search space bounds and termination criteria, both are described later). The STEP algorithm has 2 parameters: the Jones factor $\epsilon = 10^{-8}$ and the maximum interval difficulty set to 10^7 (value determined by experimenting with the Rastrigin function).

2.4 Box Constraints? Yes

Both algorithms were ordered to find the optimum in the hypercube $\langle -6, 6\rangle^D$. Why this choice if the benchmark functions have the global optimum in the interval $\langle -5, 5\rangle^D$? Because of the behaviour of the `fminbnd` function. As written in the MATLAB documentation, the fminbnd function almost never samples the boundaries of the interval which would make it extremely difficult for the algorithm to find the solution e.g. for the Linear slope function.

2.5 The Crafting Effort

For both algorithms the same setting is used for all problems, so that CrE = 0.

2.6 Invariance Properties

The algorithm searches in axis-parallel directions and thus is *not invariant with respect to rotation*. Both algorithms, `fminbnd` and STEP, make use of the fitness values when deciding where to sample a new point, thus are *not invariant with respect to order-preserving transformations of the fitness function*.

3. EXPERIMENTAL PROCEDURE

The standard experimental procedure of BBOB was used: the algorithms were run on 24 benchmark functions, 5 instances each, 3 runs per instance. Each multistart run was finished

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

Each individual launch of the basic line search algorithm was interrupted (and possibly restarted)

- if any of the above mentioned criteria was satisfied, or
- if two consecutive cycles over all directions ended up with solutions which are closer than 10^{-10}.

The individual univariate searches were stopped

- in case of `fminbnd`, when the target fitness value was reached, or the maximum allowed number of function evaluations was reached (100), or when the boundary points of the interval got closer than 10^{-10}, and
- in case of STEP, when the target fitness value was reached, or when maximum allowed number of function evaluations was reached (1000); moreover an interval was not used for further sampling if its boundary points were closer than 10^{-10}, or when the difficulty of the interval was higher than 10^7.

4. RESULTS

Results from experiments according to [2] on the benchmark functions given in [1, 3] are presented in Figures 2, 3 and 4 and in Tables 1 and 2.

In Fig. 2, we show only those functions which reveal some interesting differences between the two methods. On functions which are not shown, both functions performed similarly (similarly badly).

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 results for `fminbnd` were between $3.7 \cdot 10^{-4}$ and $3.9 \cdot 10^{-4}$ seconds per function evaluation for all the tested search space dimensionalities (2–40). For STEP line search, the results were between $5.1 \cdot 10^{-4}$ and $5.9 \cdot 10^{-4}$. 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 results for line search method are expectable: line search is unable to optimize non-separable functions. Line search method with the `fminbnd` univariate optimizer is very fast when solving quadratic separable functions (Sphere, Ellipsoid), but is unable to solve the Rastrigin function (due to multimodality the `fminbnd` gets trapped in local optimum). Line search with STEP is effective method for optimization of separable functions. It is slower (about 10 times) than `fminbnd` when optimizing univariate functions (it spends some evaluations on global search), on the other hand it is able to solve the Rastrigin and the Skew Rastrigin-Buche function (since it is a univariate global search method).

On the other hand, the line search method discovered solutions of the two Gallagher functions (or at least solutions of some of their instances) which might be an indication that these functions may be easier for algorithms that exploit separability of the problem.

Acknowledgements

The project was supported by the Ministry of Education, Youth and Sport of the Czech Republic with the grant No. MSM6840770012 entitled "Transdisciplinary Research in Biomedical Engineering II".

7. 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] S. Swarzberg, G. Seront, and H. Bersini. S.T.E.P.: The Easiest Way to Optimize a Function. In *IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 519–524 vol.1, 1994.

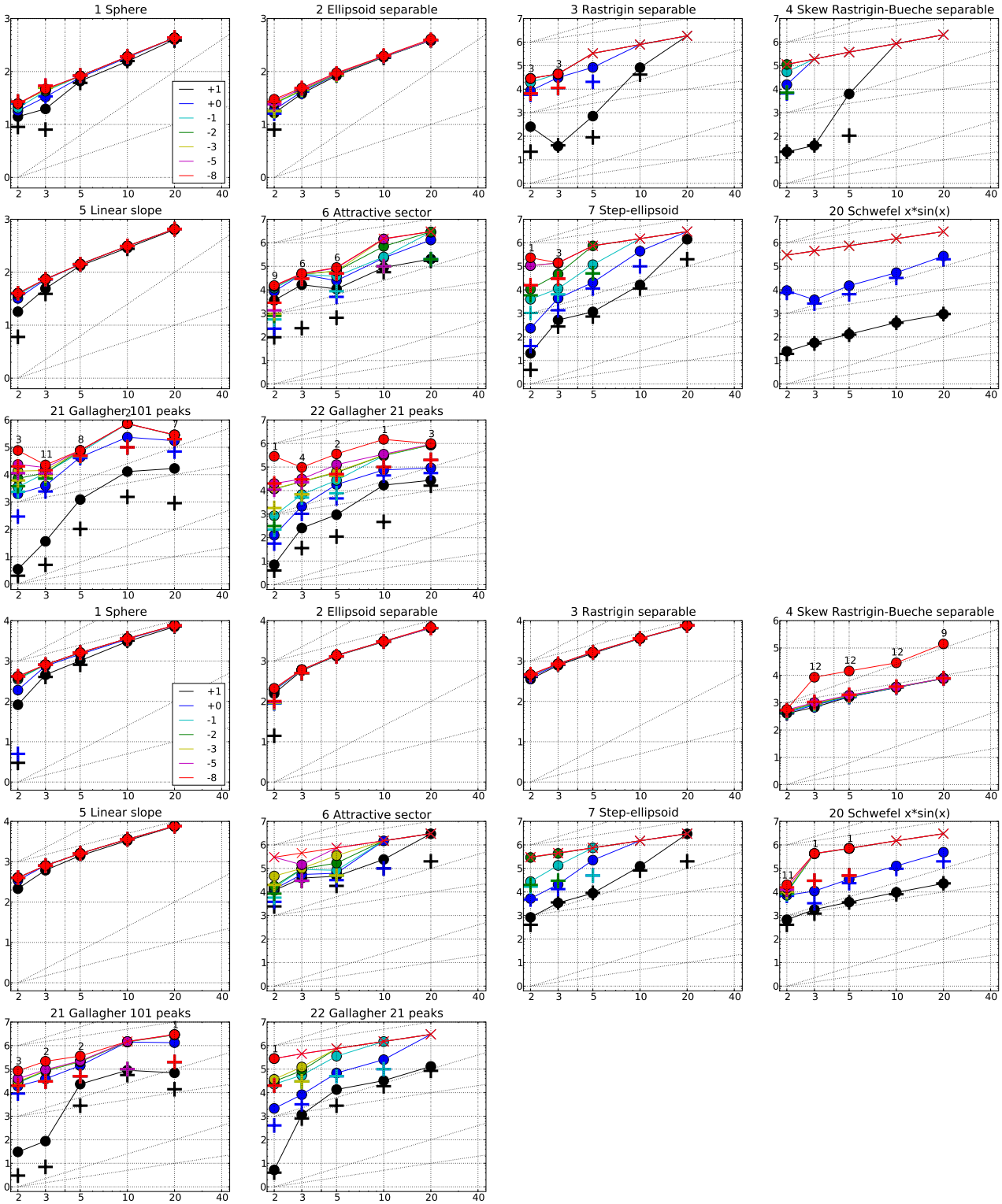

Figure 2: MATLAB fminbnd (upper 10 graphs) and STEP (lower 10 graphs): Expected Running Time (ERT, ●) to reach $f_{\mathrm{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given 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 (×) 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	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
	f_1 **in 5-D**, N=15, mFE=128					f_1 **in 20-D**, N=15, mFE=498				
10	15	6.6e1	5.7e1	7.4e1	6.6e1	15	4.0e2	3.8e2	4.2e2	4.0e2
1	15	7.7e1	7.1e1	8.5e1	7.7e1	15	4.3e2	4.1e2	4.4e2	4.3e2
1e−1	15	8.2e1	7.5e1	8.8e1	8.2e1	15	4.3e2	4.2e2	4.5e2	4.3e2
1e−3	15	8.2e1	7.6e1	8.9e1	8.2e1	15	4.3e2	4.2e2	4.5e2	4.3e2
1e−5	15	8.3e1	7.6e1	8.9e1	8.3e1	15	4.3e2	4.2e2	4.5e2	4.3e2
1e−8	15	8.3e1	7.6e1	8.9e1	8.3e1	15	4.3e2	4.2e2	4.5e2	4.3e2
	f_3 **in 5-D**, N=15, mFE=28220					f_3 **in 20-D**, N=15, mFE=135793				
10	15	7.2e2	2.4e2	1.2e3	7.2e2	0	*19e+0*	*14e+0*	*26e+0*	5.6e4
1	3	8.4e4	7.0e4	9.8e4	2.1e4	.	.	.	.	.
1e−1	0	*21e−1*	*34e−2*	*70e−1*	1.3e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_5 **in 5-D**, N=15, mFE=170					f_5 **in 20-D**, N=15, mFE=680				
10	15	1.3e2	1.3e2	1.4e2	1.3e2	15	6.4e2	6.3e2	6.4e2	6.4e2
1	15	1.4e2	1.4e2	1.4e2	1.4e2	15	6.5e2	6.5e2	6.5e2	6.5e2
1e−1	15	1.4e2	1.4e2	1.4e2	1.4e2	15	6.5e2	6.5e2	6.5e2	6.5e2
1e−3	15	1.4e2	1.4e2	1.4e2	1.4e2	15	6.5e2	6.5e2	6.5e2	6.5e2
1e−5	15	1.4e2	1.4e2	1.4e2	1.4e2	15	6.5e2	6.5e2	6.5e2	6.5e2
1e−8	15	1.4e2	1.4e2	1.4e2	1.4e2	15	6.5e2	6.5e2	6.5e2	6.5e2
	f_7 **in 5-D**, N=15, mFE=50001					f_7 **in 20-D**, N=15, mFE=200001				
10	15	1.2e3	8.5e2	1.5e3	1.2e3	2	1.4e6	1.3e6	1.5e6	1.2e5
1	12	2.1e4	1.3e4	2.8e4	1.9e4	0	*15e+0*	*94e−1*	*27e+0*	1.3e5
1e−1	5	1.2e5	9.9e4	1.3e5	4.1e4	.	.	.	.	.
1e−3	0	*40e−2*	*21e−3*	*13e−1*	2.2e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_9 **in 5-D**, N=15, mFE=50001					f_9 **in 20-D**, N=15, mFE=200001				
10	15	4.4e2	2.6e2	6.3e2	4.4e2	15	8.9e4	7.8e4	1.0e5	8.9e4
1	12	1.7e4	9.2e3	2.5e4	1.2e4	2	1.4e6	1.4e6	1.5e6	2.0e5
1e−1	11	3.9e4	3.3e4	4.5e4	2.8e4	0	*32e−1*	*89e−2*	*67e−1*	2.0e5
1e−3	0	*25e−3*	*50e−4*	*40e−1*	4.5e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{11} **in 5-D**, N=15, mFE=50001					f_{11} **in 20-D**, N=15, mFE=200001				
10	0	*61e+0*	*28e+0*	*10e+1*	4.5e4	0	*22e+1*	*19e+1*	*31e+1*	2.0e5
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{13} **in 5-D**, N=15, mFE=50000					f_{13} **in 20-D**, N=15, mFE=200000				
10	15	4.3e3	3.0e3	5.8e3	4.3e3	15	1.2e4	6.4e3	1.8e4	1.2e4
1	13	2.9e4	2.3e4	3.4e4	2.8e4	15	3.8e4	2.6e4	5.1e4	3.8e4
1e−1	5	1.4e5	1.3e5	1.5e5	4.8e4	10	1.9e5	1.5e5	2.2e5	1.3e5
1e−3	0	*19e−2*	*26e−4*	*24e−1*	3.5e4	1	2.9e6	2.9e6	3.0e6	2.0e5
1e−5	.	.	.	.	.	0	*53e−3*	*22e−4*	*23e−2*	1.1e5
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{15} **in 5-D**, N=15, mFE=50001					f_{15} **in 20-D**, N=15, mFE=200001				
10	14	1.8e4	1.3e4	2.3e4	1.7e4	0	*14e+1*	*11e+1*	*17e+1*	1.1e5
1	0	*60e−1*	*20e−1*	*91e−1*	2.2e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{17} **in 5-D**, N=15, mFE=50001					f_{17} **in 20-D**, N=15, mFE=200001				
10	15	1.2e3	5.1e1	2.4e3	1.2e3	12	6.2e4	2.9e4	1.0e5	6.2e4
1	13	1.4e4	7.7e3	2.0e4	1.2e4	0	*82e−1*	*54e−1*	*11e+0*	2.0e5
1e−1	4	1.6e5	1.4e5	1.7e5	3.6e4	.	.	.	.	.
1e−3	0	*16e−2*	*74e−3*	*16e−1*	2.2e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{19} **in 5-D**, N=15, mFE=50000					f_{19} **in 20-D**, N=15, mFE=200000				
10	15	5.4e1	3.9e1	7.0e1	5.4e1	15	1.2e3	9.2e2	1.4e3	1.2e3
1	15	3.0e3	2.1e3	3.8e3	3.0e3	0	*38e−1*	*33e−1*	*47e−1*	8.9e4
1e−1	0	*38e−2*	*22e−2*	*60e−2*	1.3e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{21} **in 5-D**, N=15, mFE=50000					f_{21} **in 20-D**, N=15, mFE=200000				
10	15	1.2e3	3.5e2	2.2e3	1.2e3	15	1.7e4	8.3e3	2.6e4	1.7e4
1	11	4.3e4	3.4e4	5.2e4	3.0e4	9	1.7e5	1.3e5	2.2e5	1.4e5
1e−1	9	6.5e4	5.6e4	7.4e4	3.9e4	7	2.8e5	2.2e5	3.3e5	1.6e5
1e−3	8	7.6e4	6.6e4	8.5e4	4.0e4	7	2.8e5	2.2e5	3.3e5	1.6e5
1e−5	8	7.6e4	6.6e4	8.6e4	4.0e4	7	2.8e5	2.3e5	3.4e5	1.6e5
1e−8	8	7.7e4	6.7e4	8.7e4	4.0e4	7	2.9e5	2.3e5	3.4e5	1.6e5
	f_{23} **in 5-D**, N=15, mFE=50000					f_{23} **in 20-D**, N=15, mFE=200000				
10	15	5.3e0	4.0e0	6.8e0	5.3e0	15	1.4e1	9.2e0	1.9e1	1.4e1
1	15	5.7e3	4.2e3	7.2e3	5.7e3	7	3.3e5	2.8e5	3.7e5	1.8e5
1e−1	0	*45e−2*	*21e−2*	*62e−2*	2.8e4	0	*10e−1*	*71e−2*	*12e−1*	1.3e5
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
	f_2 **in 5-D**, N=15, mFE=121					f_2 **in 20-D**, N=15, mFE=611				
10	15	8.3e1	8.0e1	8.7e1	8.3e1	15	3.8e2	3.8e2	3.9e2	3.8e2
1	15	8.7e1	8.4e1	9.0e1	8.7e1	15	3.9e2	3.8e2	3.9e2	3.9e2
1e−1	15	8.8e1	8.5e1	9.1e1	8.8e1	15	3.9e2	3.8e2	3.9e2	3.9e2
1e−3	15	9.0e1	8.7e1	9.3e1	9.0e1	15	3.9e2	3.8e2	4.0e2	3.9e2
1e−5	15	9.2e1	8.9e1	9.5e1	9.2e1	15	3.9e2	3.9e2	4.0e2	3.9e2
1e−8	15	9.5e1	9.1e1	9.8e1	9.5e1	15	4.1e2	3.9e2	4.3e2	4.1e2
	f_4 **in 5-D**, N=15, mFE=32641					f_4 **in 20-D**, N=15, mFE=159528				
10	12	6.3e3	2.8e3	1.0e4	4.5e3	0	*49e+0*	*32e+0*	*76e+0*	5.6e4
1	0	*42e−1*	*21e−1*	*13e+0*	1.0e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_6 **in 5-D**, N=15, mFE=50000					f_6 **in 20-D**, N=15, mFE=200000				
10	13	1.1e4	4.4e3	1.8e4	1.1e4	9	2.0e5	1.6e5	2.5e5	9.1e4
1	11	2.5e4	1.5e4	3.5e4	1.9e4	2	1.3e6	1.1e6	1.5e6	2.0e5
1e−1	9	3.8e4	2.6e4	4.9e4	1.9e4	1	2.8e6	2.6e6	3.0e6	2.0e5
1e−3	7	6.4e4	4.8e4	7.8e4	2.6e4	0	*72e−1*	*36e−2*	*42e+0*	2.0e5
1e−5	6	8.5e4	7.0e4	1.0e5	3.8e4	.	.	.	.	.
1e−8	6	8.9e4	7.4e4	1.0e5	3.9e4	.	.	.	.	.
	f_8 **in 5-D**, N=15, mFE=50001					f_8 **in 20-D**, N=15, mFE=200000				
10	15	7.7e2	4.3e2	1.1e3	7.7e2	15	1.9e4	6.4e3	3.1e4	1.9e4
1	6	7.8e4	5.9e4	9.6e4	3.5e4	5	4.4e5	3.6e5	5.2e5	1.9e5
1e−1	4	1.5e5	1.3e5	1.7e5	4.1e4	1	2.9e6	2.7e6	3.0e6	2.0e5
1e−3	0	*34e−1*	*20e−3*	*57e−1*	4.5e4	0	*40e−1*	*20e−2*	*59e−1*	2.0e5
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{10} **in 5-D**, N=15, mFE=50000					f_{10} **in 20-D**, N=15, mFE=200001				
10	0	*25e+1*	*28e+0*	*87e+1*	4.5e4	0	*22e+2*	*77e+1*	*30e+2*	2.0e5
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{12} **in 5-D**, N=15, mFE=50001					f_{12} **in 20-D**, N=15, mFE=200001				
10	9	3.4e4	2.3e4	4.5e4	2.8e4	10	1.0e5	6.2e4	1.4e5	6.1e4
1	2	3.3e5	2.8e5	3.8e5	5.0e4	3	8.0e5	6.7e5	9.3e5	6.8e4
1e−1	1	7.0e5	6.5e5	7.5e5	5.0e4	0	*76e−1*	*32e−2*	*26e+0*	2.0e5
1e−3	0	*68e−1*	*33e−2*	*45e+0*	4.5e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{14} **in 5-D**, N=15, mFE=50000					f_{14} **in 20-D**, N=15, mFE=200000				
10	15	5.7e1	4.4e1	7.2e1	5.7e1	15	6.3e2	5.8e2	6.8e2	6.3e2
1	15	1.8e2	1.5e2	2.1e2	1.8e2	15	1.2e3	1.2e3	1.3e3	1.2e3
1e−1	15	2.5e2	2.1e2	2.8e2	2.5e2	15	1.7e3	1.6e3	1.8e3	1.7e3
1e−3	15	9.7e3	7.7e3	1.2e4	9.7e3	15	5.4e4	4.7e4	6.1e4	5.4e4
1e−5	0	*35e−5*	*88e−6*	*44e−5*	4.5e4	0	*31e−5*	*23e−5*	*58e−5*	2.0e5
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{16} **in 5-D**, N=15, mFE=50001					f_{16} **in 20-D**, N=15, mFE=200001				
10	15	3.9e2	2.6e2	5.3e2	3.9e2	8	2.2e5	1.6e5	2.7e5	1.3e5
1	14	1.7e4	1.2e4	2.2e4	1.6e4	0	*95e−1*	*76e−1*	*19e+0*	1.1e5
1e−1	2	3.5e5	3.3e5	3.8e5	5.0e4	.	.	.	.	.
1e−3	0	*37e−2*	*43e−3*	*82e−2*	2.8e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{18} **in 5-D**, N=15, mFE=50000					f_{18} **in 20-D**, N=15, mFE=200001				
10	15	6.6e3	1.4e3	1.2e4	6.6e3	1	2.8e6	2.6e6	3.0e6	2.0e5
1	13	2.7e4	2.1e4	3.3e4	2.2e4	0	*26e+0*	*13e+0*	*39e+0*	1.8e5
1e−1	0	*31e−2*	*13e−2*	*14e−1*	3.2e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{20} **in 5-D**, N=15, mFE=50000					f_{20} **in 20-D**, N=15, mFE=200000				
10	15	1.3e2	1.0e2	1.6e2	1.3e2	15	9.3e2	8.7e2	9.9e2	9.3e2
1	14	1.5e4	9.9e3	2.1e4	1.3e4	8	2.7e5	2.3e5	3.1e5	1.8e5
1e−1	0	*65e−2*	*33e−2*	*87e−2*	2.5e4	0	*97e−2*	*83e−2*	*11e−1*	8.9e4
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.
	f_{22} **in 5-D**, N=15, mFE=50001					f_{22} **in 20-D**, N=15, mFE=200001				
10	15	9.4e2	5.0e2	1.4e3	9.4e2	15	2.7e4	1.9e4	3.7e4	2.7e4
1	12	1.8e4	1.0e4	2.7e4	7.5e3	13	9.2e4	6.8e4	1.2e5	8.3e4
1e−1	11	2.7e4	1.8e4	3.6e4	1.3e4	3	8.6e5	7.7e5	9.5e5	1.5e5
1e−3	8	6.3e4	5.2e4	7.4e4	3.0e4	3	8.9e5	8.1e5	9.6e5	1.6e5
1e−5	5	1.3e5	1.1e5	1.4e5	4.2e4	3	9.2e5	8.7e5	9.7e5	1.7e5
1e−8	2	3.6e5	3.5e5	3.8e5	5.0e4	3	9.9e5	9.8e5	1.0e6	2.0e5
	f_{24} **in 5-D**, N=15, mFE=50000					f_{24} **in 20-D**, N=15, mFE=200000				
10	15	1.5e4	1.2e4	1.8e4	1.5e4	0	*15e+1*	*12e+1*	*17e+1*	1.0e5
1	0	*63e−1*	*16e−1*	*94e−1*	1.6e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 1: fminbnd Line Search: 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 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 (RT_{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 2 for the names of functions.

Δf	#	ERT	10%	90%	RT_{succ}
f_1 in 5-D, N=15, mFE=1618					
10	15	1.0e3	8.6e2	1.2e3	1.0e3
1	15	1.5e3	1.4e3	1.6e3	1.5e3
1e−1	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−3	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−5	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−8	15	1.6e3	1.6e3	1.6e3	1.6e3
f_1 in 20-D, N=15, mFE=7618					
10	15	7.1e3	6.8e3	7.3e3	7.1e3
1	15	7.5e3	7.5e3	7.6e3	7.5e3
1e−1	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−3	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−5	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−8	15	7.6e3	7.6e3	7.6e3	7.6e3
f_2 in 5-D, N=15, mFE=1633					
10	15	1.3e3	1.3e3	1.4e3	1.3e3
1	15	1.4e3	1.3e3	1.4e3	1.4e3
1e−1	15	1.4e3	1.3e3	1.4e3	1.4e3
1e−3	15	1.4e3	1.3e3	1.4e3	1.4e3
1e−5	15	1.4e3	1.3e3	1.4e3	1.4e3
1e−8	15	1.4e3	1.4e3	1.4e3	1.4e3
f_2 in 20-D, N=15, mFE=9447					
10	15	6.5e3	6.5e3	6.6e3	6.5e3
1	15	6.6e3	6.5e3	6.6e3	6.6e3
1e−1	15	6.6e3	6.6e3	6.6e3	6.6e3
1e−3	15	6.6e3	6.6e3	6.6e3	6.6e3
1e−5	15	6.6e3	6.6e3	6.6e3	6.6e3
1e−8	15	6.9e3	6.7e3	7.2e3	6.9e3
f_3 in 5-D, N=15, mFE=1680					
10	15	1.6e3	1.5e3	1.6e3	1.6e3
1	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−1	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−3	15	1.6e3	1.6e3	1.7e3	1.6e3
1e−5	15	1.7e3	1.6e3	1.7e3	1.7e3
1e−8	15	1.7e3	1.7e3	1.7e3	1.7e3
f_3 in 20-D, N=15, mFE=7667					
10	15	7.5e3	7.4e3	7.6e3	7.5e3
1	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−1	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−3	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−5	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−8	15	7.7e3	7.6e3	7.7e3	7.7e3
f_4 in 5-D, N=15, mFE=50000					
10	15	1.6e3	1.6e3	1.6e3	1.6e3
1	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−1	15	1.7e3	1.7e3	1.7e3	1.7e3
1e−3	15	1.8e3	1.8e3	1.8e3	1.8e3
1e−5	15	1.9e3	1.9e3	1.9e3	1.9e3
1e−8	12	1.4e4	6.4e3	2.2e4	9.9e3
f_4 in 20-D, N=15, mFE=200000					
10	15	7.6e3	7.6e3	7.6e3	7.6e3
1	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−1	15	7.7e3	7.7e3	7.7e3	7.7e3
1e−3	15	7.7e3	7.7e3	7.7e3	7.7e3
1e−5	15	7.8e3	7.7e3	7.8e3	7.8e3
1e−8	9	1.4e5	9.8e4	1.8e5	7.2e4
f_5 in 5-D, N=15, mFE=1602					
10	15	1.4e3	1.3e3	1.5e3	1.4e3
1	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−1	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−3	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−5	15	1.6e3	1.6e3	1.6e3	1.6e3
1e−8	15	1.6e3	1.6e3	1.6e3	1.6e3
f_5 in 20-D, N=15, mFE=7602					
10	15	7.5e3	7.4e3	7.5e3	7.5e3
1	15	7.6e3	7.5e3	7.6e3	7.6e3
1e−1	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−3	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−5	15	7.6e3	7.6e3	7.6e3	7.6e3
1e−8	15	7.6e3	7.6e3	7.6e3	7.6e3
f_6 in 5-D, N=15, mFE=50008					
10	9	4.7e4	3.5e4	5.7e4	2.7e4
1	8	6.2e4	5.2e4	7.2e4	3.0e4
1e−1	7	8.3e4	7.2e4	9.5e4	3.4e4
1e−3	2	3.5e5	3.3e5	3.8e5	3.7e4
1e−5	0	*21e−2*	*59e−6*	*42e+0*	4.5e4
1e−8	.	.	.	.	.
f_6 in 20-D, N=15, mFE=200038					
10	1	3.0e6	2.9e6	3.0e6	2.0e5
1	0	*59e+0*	*11e+0*	*21e+1*	2.0e5
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_7 in 5-D, N=15, mFE=50000					
10	15	8.8e3	7.4e3	1.0e4	8.8e3
1	3	2.3e5	2.1e5	2.4e5	4.1e4
1e−1	1	7.5e5	7.5e5	7.5e5	5.0e4
1e−3	0	*19e−1*	*39e−2*	*33e−1*	3.2e4
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_7 in 20-D, N=15, mFE=200000					
10	1	3.0e6	2.9e6	3.0e6	1.5e5
1	0	*29e+0*	*16e+0*	*64e+0*	1.0e5
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_8 in 5-D, N=15, mFE=50008					
10	15	4.7e3	4.1e3	5.3e3	4.7e3
1	11	2.6e4	1.7e4	3.4e4	2.3e4
1e−1	5	1.1e5	9.3e4	1.3e5	2.7e4
1e−3	1	7.1e5	6.8e5	7.5e5	5.0e4
1e−5	0	*53e−2*	*54e−4*	*46e−1*	4.5e4
1e−8	.	.	.	.	.
f_8 in 20-D, N=15, mFE=200038					
10	15	4.9e4	4.7e4	5.1e4	4.9e4
1	5	4.8e5	4.2e5	5.3e5	1.5e5
1e−1	3	9.0e5	8.3e5	9.7e5	1.7e5
1e−3	0	*12e−1*	*74e−4*	*44e−1*	2.0e5
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_9 in 5-D, N=15, mFE=50008					
10	13	1.8e4	1.3e4	2.4e4	1.6e4
1	1	7.1e5	6.7e5	7.5e5	5.0e4
1e−1	0	*19e−1*	*12e−1*	*11e+1*	4.5e4
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_9 in 20-D, N=15, mFE=200000					
10	0	*18e+0*	*15e+0*	*73e+0*	2.0e5
1	.	.	.	.	.
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{10} in 5-D, N=15, mFE=50008					
10	0	*25e+2*	*38e+1*	*24e+3*	4.5e4
1	.	.	.	.	.
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{10} in 20-D, N=15, mFE=200038					
10	0	*18e+3*	*81e+2*	*35e+3*	2.0e5
1	.	.	.	.	.
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{11} in 5-D, N=15, mFE=50008					
10	1	7.0e5	6.5e5	7.5e5	5.0e4
1	0	*73e+0*	*14e+0*	*17e+1*	4.5e4
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{11} in 20-D, N=15, mFE=200038					
10	0	*29e+1*	*21e+1*	*40e+1*	2.0e5
1	.	.	.	.	.
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{12} in 5-D, N=15, mFE=50008					
10	8	5.0e4	3.7e4	6.3e4	2.3e4
1	2	3.3e5	2.9e5	3.8e5	5.0e4
1e−1	0	*57e−1*	*68e−2*	*40e+0*	4.5e4
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{12} in 20-D, N=15, mFE=200036					
10	7	2.4e5	1.8e5	3.1e5	9.1e4
1	2	1.3e6	1.1e6	1.5e6	2.0e5
1e−1	0	*16e+0*	*90e−2*	*53e+0*	2.0e5
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{13} in 5-D, N=15, mFE=50000					
10	7	7.2e4	6.0e4	8.6e4	3.2e4
1	3	2.2e5	2.0e5	2.4e5	5.0e4
1e−1	1	7.2e5	6.9e5	7.5e5	5.0e4
1e−3	0	*22e+0*	*26e−2*	*56e+0*	3.5e4
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{13} in 20-D, N=15, mFE=200000					
10	7	3.0e5	2.6e5	3.5e5	1.2e5
1	1	2.9e6	2.8e6	3.0e6	2.0e5
1e−1	0	*12e+0*	*15e−1*	*59e+0*	1.8e5
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{14} in 5-D, N=15, mFE=50000					
10	15	1.1e3	9.1e2	1.4e3	1.1e3
1	15	3.9e3	3.3e3	4.5e3	3.9e3
1e−1	15	5.6e3	5.0e3	6.2e3	5.6e3
1e−3	0	*35e−4*	*19e−4*	*56e−4*	4.5e4
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{14} in 20-D, N=15, mFE=200000					
10	15	1.4e4	1.2e4	1.6e4	1.4e4
1	15	2.8e4	2.6e4	2.9e4	2.8e4
1e−1	15	3.6e4	3.4e4	3.8e4	3.6e4
1e−3	0	*46e−4*	*28e−4*	*62e−4*	2.0e5
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{15} in 5-D, N=15, mFE=50000					
10	1	7.1e5	6.6e5	7.5e5	5.0e4
1	1	7.5e5	7.4e5	7.5e5	5.0e4
1e−1	0	*24e+0*	*14e+0*	*47e+0*	4.5e4
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{15} in 20-D, N=15, mFE=200000					
10	0	*22e+1*	*16e+1*	*37e+1*	2.0e5
1	.	.	.	.	.
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{16} in 5-D, N=15, mFE=50000					
10	15	1.7e3	1.0e3	2.3e3	1.7e3
1	4	1.7e5	1.6e5	1.8e5	5.0e4
1e−1	0	*13e−1*	*21e−2*	*29e−1*	3.5e4
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{16} in 20-D, N=15, mFE=200000					
10	7	3.3e5	2.9e5	3.7e5	1.4e5
1	0	*10e+0*	*76e−1*	*16e+0*	1.0e5
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{17} in 5-D, N=15, mFE=50000					
10	15	7.9e2	5.5e2	1.0e3	7.9e2
1	4	1.5e5	1.2e5	1.7e5	3.0e4
1e−1	0	*26e−1*	*38e−2*	*92e−1*	4.5e4
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{17} in 20-D, N=15, mFE=200000					
10	10	1.1e5	6.8e4	1.5e5	6.5e4
1	0	*78e−1*	*59e−1*	*15e+0*	2.0e5
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{18} in 5-D, N=15, mFE=50000					
10	13	1.5e4	8.6e3	2.2e4	1.5e4
1	4	1.5e5	1.3e5	1.7e5	4.0e4
1e−1	2	3.4e5	3.1e5	3.8e5	5.0e4
1e−3	0	*60e−1*	*68e−3*	*11e+0*	4.5e4
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{18} in 20-D, N=15, mFE=200000					
10	0	*31e+0*	*20e+0*	*64e+0*	2.0e5
1	.	.	.	.	.
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{19} in 5-D, N=15, mFE=50000					
10	15	9.1e2	7.7e2	1.1e3	9.1e2
1	15	9.5e3	6.3e3	1.3e4	9.5e3
1e−1	2	3.5e5	3.3e5	3.8e5	5.0e4
1e−3	0	*23e−2*	*90e−3*	*64e−2*	4.0e4
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{19} in 20-D, N=15, mFE=200000					
10	15	7.8e3	7.2e3	8.5e3	7.8e3
1	0	*41e−1*	*31e−1*	*54e−1*	1.1e5
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{20} in 5-D, N=15, mFE=50000					
10	15	3.7e3	3.2e3	4.2e3	3.7e3
1	11	3.5e4	2.6e4	4.4e4	2.5e4
1e−1	1	7.0e5	6.6e5	7.5e5	5.0e4
1e−3	1	7.1e5	6.6e5	7.5e5	5.0e4
1e−5	1	7.1e5	6.7e5	7.5e5	5.0e4
1e−8	1	7.2e5	6.9e5	7.5e5	5.0e4
f_{20} in 20-D, N=15, mFE=200000					
10	15	2.3e4	2.2e4	2.4e4	2.3e4
1	5	4.9e5	4.3e5	5.5e5	1.5e5
1e−1	0	*10e−1*	*80e−2*	*12e−1*	2.0e5
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{21} in 5-D, N=15, mFE=50000					
10	11	2.3e4	1.3e4	3.3e4	1.3e4
1	4	1.4e5	1.2e5	1.6e5	2.7e4
1e−1	3	2.1e5	1.8e5	2.4e5	5.0e4
1e−3	3	2.1e5	1.9e5	2.4e5	5.0e4
1e−5	3	2.2e5	2.1e5	2.4e5	5.0e4
1e−8	2	3.6e5	3.4e5	3.8e5	5.0e4
f_{21} in 20-D, N=15, mFE=200000					
10	12	7.0e4	3.6e4	1.0e5	6.7e4
1	2	1.3e6	1.2e6	1.5e6	2.0e5
1e−1	1	2.8e6	2.7e6	3.0e6	2.0e5
1e−3	1	2.9e6	2.8e6	3.0e6	2.0e5
1e−5	1	2.9e6	2.8e6	3.0e6	2.0e5
1e−8	1	3.0e6	3.0e6	3.0e6	2.0e5
f_{22} in 5-D, N=15, mFE=50000					
10	13	1.4e4	7.3e3	2.0e4	7.6e3
1	7	6.9e4	5.4e4	8.2e4	3.1e4
1e−1	2	3.6e5	3.4e5	3.8e5	5.0e4
1e−3	0	*11e−1*	*31e−3*	*15e+0*	4.5e4
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{22} in 20-D, N=15, mFE=200000					
10	11	1.3e5	9.4e4	1.6e5	1.0e5
1	0	*51e−1*	*20e−1*	*18e+0*	2.0e5
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{23} in 5-D, N=15, mFE=50000					
10	15	4.3e0	3.3e0	5.3e0	4.3e0
1	15	3.4e3	2.4e3	4.5e3	3.4e3
1e−1	1	7.3e5	7.2e5	7.5e5	5.0e4
1e−3	0	*31e−2*	*15e−2*	*51e−2*	2.5e4
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{23} in 20-D, N=15, mFE=200000					
10	15	7.1e0	6.0e0	8.1e0	7.1e0
1	13	1.3e5	1.1e5	1.5e5	1.1e5
1e−1	0	*91e−2*	*68e−2*	*10e−1*	1.3e5
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{24} in 5-D, N=15, mFE=50000					
10	2	3.3e5	2.8e5	3.8e5	5.0e4
1	0	*15e+0*	*63e−1*	*22e+0*	4.5e4
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.
f_{24} in 20-D, N=15, mFE=200000					
10	0	*19e+1*	*95e+0*	*30e+1*	2.0e5
1	.	.	.	.	.
1e−1	.	.	.	.	.
1e−3	.	.	.	.	.
1e−5	.	.	.	.	.
1e−8	.	.	.	.	.

Table 2: STEP Line Search: 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 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 (RT_{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 2 for the names of functions.