Comparison of Multistart Global Optimization Algorithms on the BBOB Noiseless Testbed

László Pál
Sapientia - Hungarian University of Transylvania
530104 Miercurea-Ciuc, Piata Libertatii, Nr. 1, Romania
pallaszlo@sapientia.siculorum.ro

ABSTRACT

Multi Level Single Linkage is a multistart, stochastic global optimization method which relies on random sampling and local search. In this paper, we benchmarked three variants of the MLSL algorithm by using two gradient based and a derivative-free local search method on the noiseless function testbed. The three methods were also compared with a commercial multistart solver, called OQNLP (OptQuest/NLP).

Our experiment showed that, the results may be influenced essentially by the applied local search procedure. Depending of the type of the problem the gradient based local search methods are faster in the initial stage of the optimization, while the derivative-free method show a superior performance in the final phase for moderate dimensions. Considering the percentage of the solved problems, OQNLP is similar or even better (for multi-modal and weakly structured functions) in 5-D than the MLSL method equipped with the gradient type local search methods, while on 20-D the latter algorithms are usually more faster.

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

Keywords

Benchmarking, Black-box optimization, Multi level methods, Multistart heuristic, Scatter search

1. INTRODUCTION

Multistart global optimization algorithms were introduced in the 1980s for bound constrained optimization problems. Two important multistart type methods are the Clustering [1] and Multi Level Single Linkage (MLSL) [9] algorithms. The basic idea behind these methods is to form groups (clusters) of points around the local minimizers from a uniform sampled domain and start local searches no more than once in each of those groups.

The aim of the paper is to compare three variants of the MLSL method using the COCO framework [3] with OQNLP (OptQuest/NLP) [11], an other well known commercial multistart type algorithm.

The rest of this article is organized as follows. Section 2 reviews the MLSL and OQNLP algorithms. In Section 3, we describe the experiment procedure together with the algorithms parameter settings. The results are presented in Section 4 and discussed in Section 5. Section 6 concludes the paper and points out some directions for future work.

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.

2. ALGORITHMS

Multi Level Single Linkage (MLSL) has two phases: a global and a local one. The global phase consists of sampling, while the local phase is based on local searches. The local minimizer points are found by means of a local search procedure (LS), starting from appropriately chosen points from the sample drawn uniformly within the set of feasibility. A local search procedure is applied to every sample point from the reduced sample, except if there is another sample point within some critical distance r_k (defined in [9]), which has a lower function value (see Algorithm 1). The reduced sample consists of the γkN best points ($0<\gamma \leq 1$) from the cumulated sample $x_1, \ldots, x_{kN}$.

The local search method is an essential part of the MLSL. Depending on the applied local search procedure the quality of the found solution may vary significantly. Thus in this study we tested three MLSL variants by applying two gradient based and a derivative-free local search method (more details in Sec. 3).

OQNLP is a solver designed to find global optima of smooth constrained nonlinear problems. It is a multistart heuristic method which runs a local search from a variety of starting points in order to find a global minimum, or multiple local minima. The solver uses a scatter-search mechanism for generating start points. The solver steps are presented in the Algorithm 2. After an initial call to LS at the user-provided initial point, x_0, N_1 trial points are generated (Stage 1). The best point is chosen as the starting point for the next call to LS. In Stage 2, N_2 iterations are performed in which candidate starting points are generated and LS is

Algorithm 1: The MLSL algorithm

$X^* \leftarrow \emptyset$; $k \leftarrow 0$
repeat
  $k \leftarrow k + 1$
  Generate N points $x_{(k-1)N+1}, \ldots, x_{kN}$ with uniform distribution on X.
  Determine the reduced sample (X_r) consisting of the γkN best points from the cumulated sample $x_1, \ldots, x_{kN}$.
  for $i \leftarrow 1$ **to** length(X_r) **do**
    if NOT (there is such a j that $f(x_j) < f(x_i)$ and $\|x_j - x_i\| < r_k$) **then**
      Start a local search method (LS) from x_i.
      $x^* \leftarrow LS(x_i)$
      $X^* \leftarrow X^* \cup \{x^*\}$
until Some global stopping rule is satisfied.
return The smallest local minimum value found.

Algorithm 2: The OQNLP solver steps

Stage 1:
Set x_0, user initial point.
Start a local search method LS from x_0.
Generate N_1 trial points using the scatter-search mechanism on the domain X.
Start a local search from the best trial point among the N_1 points.
Initialize the regions of attraction, counters, threshold.
Stage 2:
for $i \leftarrow 1$ **to** N_2 **do**
  Generate a new trial point x_i.
  Start LS from x_i if passes the distance and merit filter tests.
return The smallest local minimum value found.

started at any one which passes the distance and merit filter tests.

The distance filter helps insure that the starting points for LS are diverse, in the sense that they are not too close to any previously found local solution. Its goal is to prevent LS from starting more than once within the basin of attraction of any local optimum.

Based on some recent comparative studies [10] on bound constrained problems, OQNLP show superior performance in terms of refining a near-optimal solution.

In our comparisons we used the commercial optimization software TOMLAB/OQNLP [5].

3. EXPERIMENT DESIGN

The main purpose of the experiment is to investigate the impact of the different local search algorithms on the MLSL method and to compare the results with those obtained by the OQNLP solver. For this reason we fixed the parameters of the MLSL algorithm to specific values and alternated the local searches.

Each of the algorithms was run on 15 instances of all the 24 functions in dimensions 2, 3, 5, 10, and 20. The maximal evaluations budget (for the MLSL) was set to $2 \cdot 10^4 D$ for each run.

MLSL has four parameters to set: the number of sample points in an iteration, the size of the reduced sample, the maximum number of function evaluations for local search, and the used local search procedure. The sample was generated from a Sobol quasi-random sequence [6] and its size was set to $50D$. From the actual sample only the best $5D$ points are considered for further analysis.

We benchmarked three variants of the MLSL algorithm by using 2 gradient type and a derivative-free local search method. The garadient based methods are the following: a quasi-Newton type (`fminunc`) and an interior point (`fmincon`) procedure from MATLAB. The first is a well-known quasi-Newton method which approximates the Hessian by the BFGS formula, while the second is an interior-point algorithm for constrained nonlinear problems. The third local search algorithm is the Nelder-Mead [7] simplex method which belongs to the class of direct search methods. All the three version of the algorithms were run on the whole testbed in all dimensions. The maximum number of function evaluations for local search was set to 10% of the total budget while the termination tolerance parameter value was set to 10^{-12}.

In the case of the OQNLP method, we used the default parameters (see in [5]) except the iteration limit which was set to $300D$. Using this limit we get approximately the same maximal budget as in the case of MLSL. Furthermore it is important that OQNLP changes its search strategy depending on the iteration limit. The local search used by OQNLP is the LSGRG2, a generalized gradient projection method.

4. RESULTS

Results from experiments according to [3] on the benchmark functions given in [2, 4] are presented in Figures 1, 2 and 3 and in Tables 1 and 2. The **expected running time (ERT)**, used in the figures and table, depends on a given target function value, $f_t = f_{opt} + \Delta f$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not reach f_t, summed over all trials and divided by the number of trials that actually reached f_t [3, 8]. **Statistical significance** is tested with the rank-sum test for a given target Δf_t (10^{-8} as in Figure 1) using, for each trial, either the number of needed function evaluations to reach Δf_t (inverted and multiplied by -1), or, if the target was not reached, the best Δf-value achieved, measured only up to the smallest number of overall function evaluations for any unsuccessful trial under consideration.

4.1 CPU Timing Experiments

The timing experiments were carried out with f_8 on a machine with Intel Dual-Core processor, 2.6 Ghz, with 2 GB RAM, on Windows 7 64bit in MATLAB R2011b 64bit. The average time per function evaluation in 2, 3, 5, 10, 20, 40 dimensions was about 13, 9.4, 7.1, 5.2, 3.9, 3.7×10^{-4} s for `fmincon`, about 6.1, 5.5, 3.9, 3.1, 2.9, 2.7×10^{-4} s for `fminunc`, about 4.5, 3.3, 2.9, 3.3, 4.6, 8.8×10^{-4} s for `simplex`, and about 8.1, 7.6, 5.7, 4.1, 3.9, 3.1×10^{-4} s for OQNLP.

5. DISCUSSION

Although the MLSL method cannot find the final solution in many cases, our aim was to reveal the differences between

the applied local search methods during the different stages of the optimization.

Considering the ERT numbers in different dimensions, we can state that the gradient type methods are usually more faster than the `simplex` method. Nevertheless there are situations when the latter method is significantly better in lower dimensions than the other methods. Such cases can be observed on the f_7, f_{10}, f_{11}, f_{13}, f_{14}, f_{16}, and f_{23} functions (see Figure 1). The OQNLP solver is faster than the MLSL method with `fmincon` and `fminunc` on the f_5, f_7, f_{20}, and f_{24} functions. On f_{24} OQNLP is even faster than the best BBOB-2009 algorithm for 2, 3, and 5 dimensions.

Regarding the proportion of solved instances, the general aspect is that the gradient type methods are faster on the initial phase of the optimization, while the derivative-free `simplex` method provides a better performance in the final stage for 2, 3, and 5 dimensions.

Considering all functions aggregated in 5-D (see Figure 2), the proportion of the solved problems by the algorithms varies between 62% and 78%. `fminunc` is the fastest for #FEs $< 100D$, while between $100D$ and $1000D$ the `fmincon` solves the largest proportion of problems. After $1000D$ evaluations the `simplex` method becomes the leader by solving 78% of the problems up to the final budget. This behavior is more pronounced on the ill-conditioned functions subgroup. For #FEs $< 200D$, the `fmincon` is the best algorithm solving more than 60% of the problems, followed by `fminunc`, OQNLP and `simplex` solving 50%, 45% and 8% of the problems. For #FEs $> 700D$, the `simplex` becomes the best competitor by solving 100% of the problems up to the final budget. This huge progress is due to the robustness of the method on the f_{10},f_{11},f_{12},f_{13} and f_{14} functions. The OQNLP is slightly faster than the `simplex` algorithm on the multi-modal and weakly structured functions. This behavior is caused by the success of the OQNLP method on the f_{19} and f_{24} functions.

In the 20-D space (see Figure 3), the previously observed advantageous properties of the `simplex` method cannot be further observed. The largest proportion of solved problems by `simplex` is about 22% on the separable functions subgroup, while on the moderate group is the lowest (about 2%). Considering all functions aggregated, `fmincon` is the fastest by solving about 58% of the problems, followed by `fminunc`, OQNLP, and `simplex` solving 55%, 50% and 15% of the problems. The previous ranking of the algorithms can be observed for the other function groups too.

6. CONCLUSIONS

We benchmarked three variants of the MLSL algorithm by using two gradient based and a derivative-free local search method on the noiseless function testbed. The three methods were also compared with OQNLP (OptQuest/NLP), a heuristic, multistart solver.

The results show that depending of the type of the problem, the gradient based local search methods are faster in the initial stage of the optimization, while the derivative-free method show a superior performance in the final phase for moderate dimensions. Considering the percentage of the solved problems, OQNLP is similar or even better (for multimodal and weakly structured functions) in 5-D than the MLSL method equipped with the gradient type local search methods, while on 20-D the latter algorithms are usually more faster.

As a feature work we propose a strategy which tries to automatically select the best local search algorithm during the optimization.

Acknowledgements

This work was supported by the Sapientia Foundation - Institute for Scientific Research with the grant No. 101/9/2013.

7. REFERENCES

[1] C. G. E. Boender, A. H. G. Rinnooy Kan, G. T. Timmer, and L. Stougie. A stochastic method for global optimization. *Mathematical Programming*, 22:125–140, 1982.

[2] 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. Updated February 2010.

[3] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2012: Experimental setup. Technical report, INRIA, 2012.

[4] 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. Updated February 2010.

[5] K. Holmström, A. O. Göran, and M. M. Edvall. User's Guide for TOMLAB/OQNLP. Tomlab Optimization, 2007.

[6] H. S. Hong and F. J. Hickernell. Algorithm 823: Implementing Scrambled Digital Sequences. *ACM Transactions on Mathematical Software*, 29:95–109, 2003.

[7] J. Nelder and R. Mead. The downhill simplex method. *Computer Journal*, 7:308–313, 1965.

[8] K. Price. Differential evolution vs. the functions of the second ICEO. In *Proceedings of the IEEE International Congress on Evolutionary Computation*, pages 153–157, 1997.

[9] A. H. G. Rinnooy Kan and G. T. Timmer. Stochastic global optimization methods part II: Multi level methods. *Mathematical Programming*, 39:57–78, 1987.

[10] L. M. Rios and N. V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 19(3):1–47, 2012.

[11] Z. Ugray, L. Lasdon, J. Plummer, R. Glover, J. Kelly, and R. Marti. Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization. *INFORMS Journal on Computing*, 19(3):328–340, 2007.

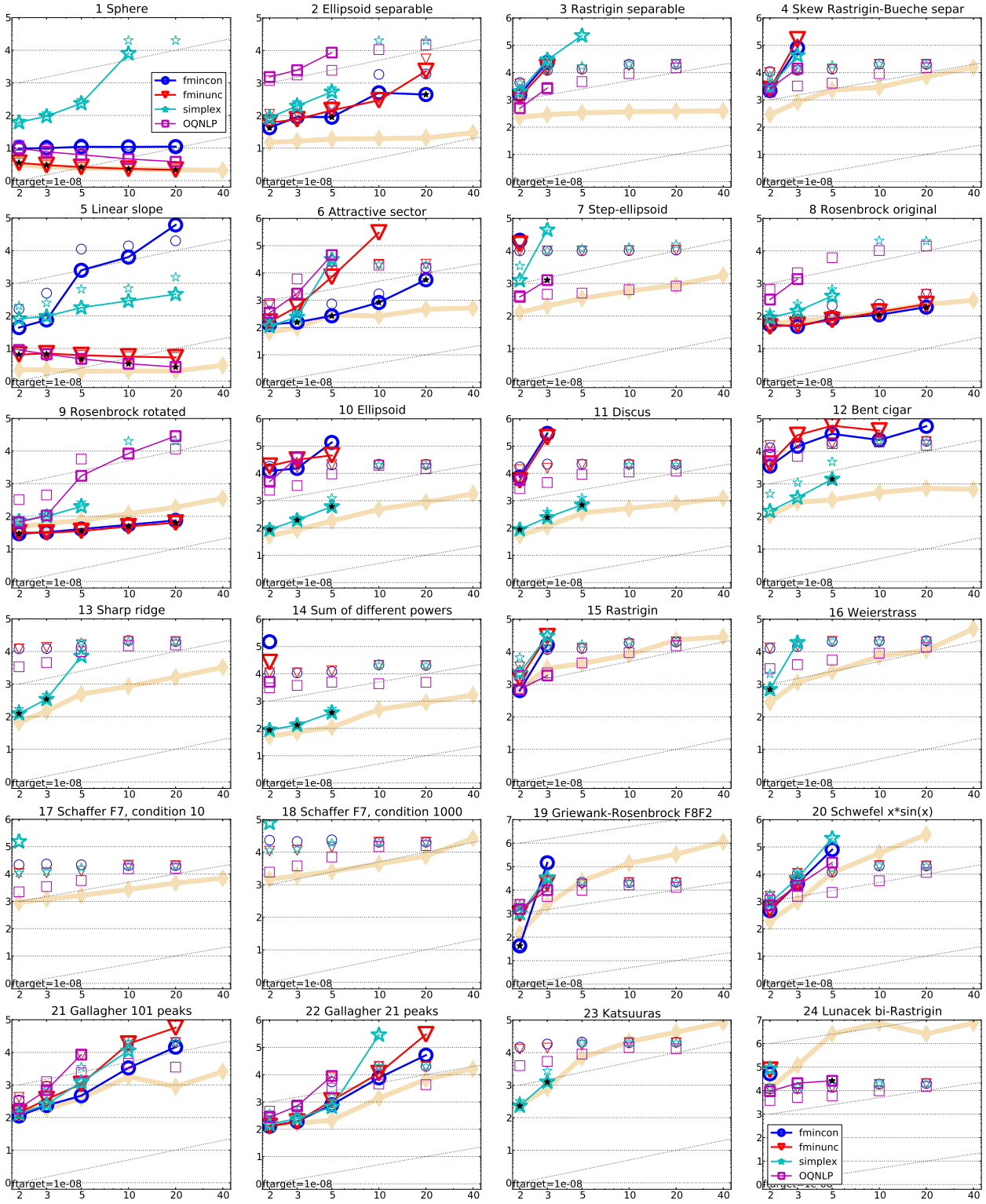

Figure 1: Expected running time (ERT in number of f-evaluations) divided by dimension for target function value 10^{-8} as $\log_{10}$ values versus dimension. Different symbols correspond to different algorithms given in the legend of f_1 and f_{24}. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Horizontal lines give linear scaling, slanted dotted lines give quadratic scaling. Black stars indicate statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: ○:fmincon, ▽:fminunc, ⋆:simplex, □:OQNLP

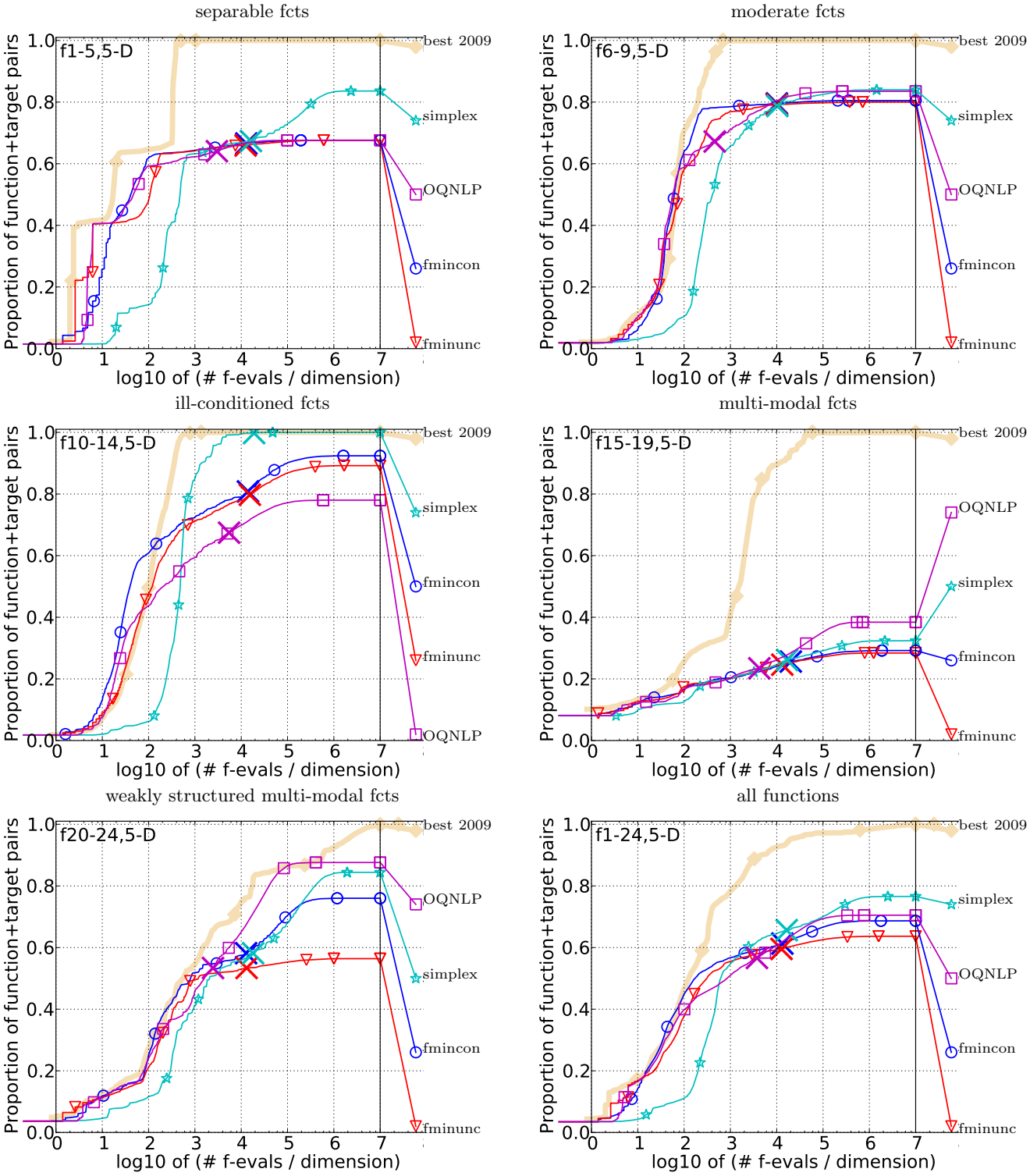

Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 5-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single target.

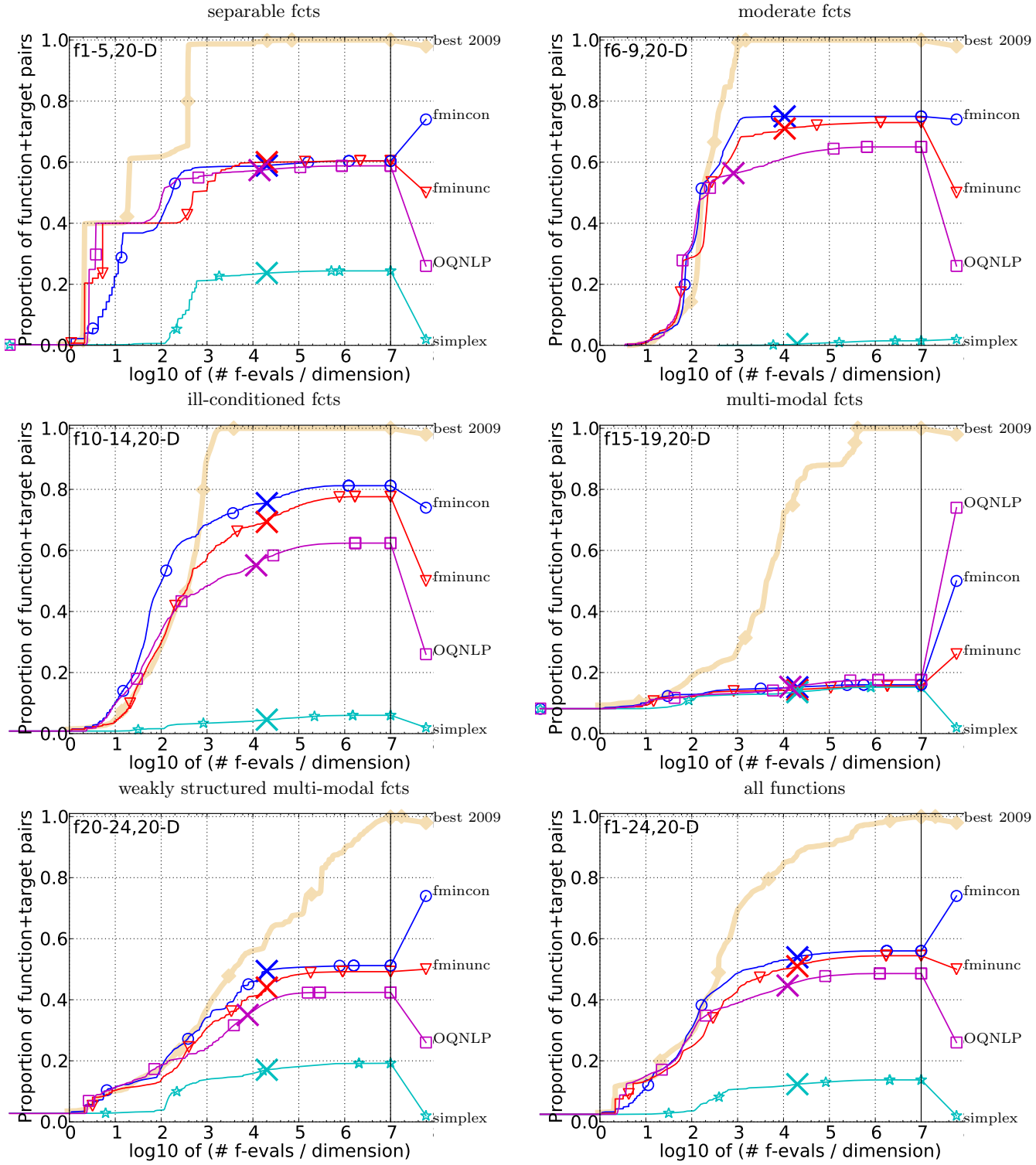

Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 20-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single target.

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f1	11	12	12	12	12	12	15/15
fmincon	**0.71**(0.3)	1.4(0.5)	2.0(0)	2.6(0.2)	3.4(0.2)	3.9(0.5)	15/15
fminunc	1.0(0.3)	**1.1**(0)	**1.1**(0)$^{\star 4}$	**1.1**(0)$^{\star 4}$	**1.1**(0)$^{\star 4}$	**1.1**(0)$^{\star 4}$	15/15
simplex	21(37)	63(27)	76(22)	85(11)	90(11)	94(12)	15/15
OQNLP	1.7(0.1)	1.7(0.1)	2.2(0.3)	2.5(0.1)	2.5(0.0)	2.6(0.0)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f2	83	87	88	90	92	94	15/15
fmincon	**1.7**(1)	**1.8**(0.9)	**1.9**(1.0)	**2.4**(1)	3.1(2)	**4.0**(2)$^{\star 2}$	15/15
fminunc	4.5(2)	5.8(2)	6.6(2)	7.1(2)	7.4(2)	7.5(2)	15/15
simplex	19(12)	25(9)	27(4)	29(3)	28(3)	28(3)	15/15
OQNLP	2.0(0.8)	2.5(1)	2.6(1)	2.8(1)	**3.0**(1)	131(151)	3/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f3	716	1622	1637	1646	1650	1654	15/15
fmincon	**5.2**(6)	122(144)	∞	∞	∞	∞ *6e4*	0/15
fminunc	5.3(5)	253(298)	∞	∞	∞	∞ *6e4*	0/15
simplex	12(7)	698(763)	**692**(798)	**688**(792)	**686**(753)	**685**(750)	1/15
OQNLP	8.6(12)	**41**(48)	∞	∞	∞	∞ *1e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f4	809	1633	1688	1817	1886	1903	15/15
fmincon	**12**(10)	∞	∞	∞	∞	∞ *7e4*	0/15
fminunc	27(43)	∞	∞	∞	∞	∞ *6e4*	0/15
simplex	39(34)	∞	∞	∞	∞	∞ *8e4*	0/15
OQNLP	25(28)	∞	∞	∞	∞	∞ *2e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f5	10	10	10	10	10	10	15/15
fmincon	2.5(0)	4.3(0)	5.5(0)	6.1(0)	6.7(0)	25(16)	13/15
fminunc	**1.9**(0)$^{\star 4}$	2.5(0)	3.1(0)	3.1(0)	3.1(0)	3.1(0)	15/15
simplex	74(77)	89(97)	90(97)	90(97)	90(97)	90(97)	15/15
OQNLP	2.3(0)	**2.4**(0)$^{\star 4}$	**2.4**(0)$^{\star 4}$	**2.4**(0)$^{\star 4}$	**2.4**(0)$^{\star 4}$	**2.4**(0)$^{\star 4}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f6	114	214	281	580	1038	1332	15/15
fmincon	1.2(0.5)	1.1(0.6)	1.2(0.6)	**0.85**(0.4)	**0.67**(0.3)*	**0.68**(0.2)$^{\star 4}_{\downarrow}$	15/15
fminunc	1.8(2)	2.2(1)	2.6(0.9)	2.1(0.5)	1.8(0.8)	6.4(9)	14/15
simplex	15(7)	13(6)	13(7)	16(16)	20(21)	43(48)	7/15
OQNLP	**0.98**(0.7)	**1.00**(0.5)	**1.00**(0.4)	2.3(2)	7.3(14)	20(22)	2/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f7	24	324	1171	1572	1572	1597	15/15
fmincon	61(73)	1086(1119)	∞	∞	∞	∞ *5e4*	0/15
fminunc	55(65)	∞	∞	∞	∞	∞ *5e4*	0/15
simplex	45(39)	155(134)	648(791)	∞	∞	∞ *5e4*	0/15
OQNLP	**26**(28)	**48**(55)	**28**(31)	∞	∞	∞ *2289*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f8	73	273	336	391	410	422	15/15
fmincon	1.0(0.3)	1.1(1)	1.0(0.9)	0.97(0.7)	0.99(0.7)	0.98(0.7)	15/15
fminunc	1.2(0.6)	0.90(0.3)	0.87(0.3)	0.87(0.2)	0.88(0.2)	**0.88**(0.2)	15/15
simplex	8.8(5)	4.9(3)	4.7(2)	4.7(2)	4.6(2)	4.8(2)	15/15
OQNLP	**0.89**(0.4)	**0.71**(0.3)	**0.75**(0.2)	**0.78**(0.2)	**0.78**(0.2)	27(33)	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f9	35	127	214	300	335	369	15/15
fmincon	0.94(0)	0.64(0.1)$_{\downarrow 4}$	0.61(0.1)$_{\downarrow 4}$	0.57(0.0)$_{\downarrow 4}$	0.56(0.0)$_{\downarrow 4}$	0.54(0.0)$_{\downarrow 4}$	15/15
fminunc	**0.55**(0)$^{\star 4}$	**0.25**(0)$^{\star 4}_{\downarrow 4}$	**0.50**(0.0)$_{\downarrow 4}$	**0.50**(0.0)$^{\star 2}_{\downarrow 4}$	**0.49**(0.0)$^{\star 3}_{\downarrow 4}$	**0.47**(0.0)$^{\star 4}_{\downarrow 4}$	15/15
simplex	5.1(3)	3.8(2)	3.6(0.9)	2.9(0.7)	2.8(0.6)	2.6(0.5)	15/15
OQNLP	0.71(0.0)	0.51(8e-3)$_{\downarrow 4}$	0.52(7e-3)$_{\downarrow 4}$	0.54(1e-2)$_{\downarrow 4}$	0.53(0.0)$_{\downarrow 4}$	1.3(0.0)	10/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f10	349	500	574	626	829	880	15/15
fmincon	**0.27**(0.1)$_{\downarrow 4}$	**0.22**(0.0)$_{\downarrow 4}$	**0.21**(0.0)$_{\downarrow 4}$	1.8(0.3)	19(20)	150(179)	2/15
fminunc	0.99(0.5)	1.0(0.3)	1.0(0.3)	**1.1**(0.4)	**2.4**(2)	92(124)	5/15
simplex	5.5(2)	4.5(2)	4.3(0.8)	4.3(0.6)	3.3(0.4)	**3.2**(0.4)$^{\star 2}$	15/15
OQNLP	0.38(0.3)$_{\downarrow 3}$	0.41(0.3)	0.89(2)	6.5(6)	120(133)	∞ *2e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f11	143	202	763	1177	1467	1673	15/15
fmincon	**0.24**(0.1)$^{\star 3}_{\downarrow 4}$	**0.22**(0.1)$^{\star 3}_{\downarrow 4}$	**0.07**(0.0)$^{\star 3}_{\downarrow 4}$	**1.1**(3)*	59(75)	∞ *1e5*	0/15
fminunc	0.90(0.4)	1.7(2)	1.5(1)	19(24)	288(304)	∞ *1e5*	0/15
simplex	8.5(6)	11(5)	4.0(1)	2.8(2)	**2.3**(1)$^{\star 2}$	**2.1**(1)$^{\star 4}$	15/15
OQNLP	0.41(0.1)$_{\downarrow 4}$	0.95(1)	0.53(0.9)	4.7(2)	142(162)	∞ *3e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f12	108	268	371	461	1303	1494	15/15
fmincon	1.3(0.4)	**0.95**(0.5)	**0.90**(0.7)	**0.93**(0.7)	**1.0**(0.9)	29(48)	5/15
fminunc	**1.0**(1.0)	1.1(1.0)	1.2(0.9)	1.5(1)	2.4(4)	80(109)	3/15
simplex	11(7)	6.7(3)	8.3(4)	8.7(11)	4.6(7)	**4.2**(6)	15/15
OQNLP	1.2(1)	0.95(0.8)	1.2(1)	20(27)	21(32)	328(399)	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f13	132	195	250	1310	1752	2255	15/15
fmincon	**0.74**(0.1)$^{\star}_{\downarrow 2}$	**0.76**(0.1)$_{\downarrow 4}$	**0.80**(0.1)$_{\downarrow 4}$	**4.1**(1)	61(79)	∞ *7e4*	0/15
fminunc	0.92(0.2)	1(0.2)	0.98(0.1)	4.9(4)	167(198)	∞ *7e4*	0/15
simplex	15(15)	17(15)	17(14)	5.0(4)	**5.1**(4)	**10**(10)	13/15
OQNLP	0.95(0.4)	0.89(0.2)	0.83(0.2)	8.4(10)	∞	∞ *3e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f14	10	41	58	139	251	476	15/15
fmincon	**0.68**(0.4)	**0.53**(0.2)$_{\downarrow 3}$	**0.64**(0.2)$^{\star 2}_{\downarrow 3}$	**0.67**(0.2)$_{\downarrow 2}$	**0.72**(0.1)	797(979)	0/15
fminunc	0.71(0.6)	0.85(0.6)	1.1(0.4)	0.89(0.3)	0.84(0.2)	1711(1988)	0/15
simplex	6.4(4)	15(11)	17(6)	8.9(2)	5.6(1)	**3.8**(1)$^{\star 2}$	15/15
OQNLP	1.4(0.9)	0.94(0.3)	1.0(0.2)	0.80(0.2)$_{\downarrow}$	7.0(9)	∞ *2e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f15	511	9310	19369	20073	20769	21359	14/15
fmincon	**8.2**(9)	**45**(56)	∞	∞	∞	∞ *6e4*	0/15
fminunc	8.8(12)	∞	∞	∞	∞	∞ *6e4*	0/15
simplex	20(21)	122(119)	∞	∞	∞	∞ *8e4*	0/15
OQNLP	12(17)	∞	∞	∞	∞	∞ *1e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f16	120	612	2663	10449	11644	12095	15/15
fmincon	**4.1**(2)	61(53)	495(553)	∞	∞	∞ *9e4*	0/15
fminunc	12(12)	154(164)	∞	∞	∞	∞ *1e5*	0/15
simplex	6.3(1)	**20**(18)*	**44**(38)	∞	∞	∞ *1e5*	0/15
OQNLP	12(19)	243(275)	118(130)	∞	∞	∞ *2e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f17	5.2	215	899	3669	6351	7934	15/15
fmincon	20(25)	172(167)	∞	∞	∞	∞ *1e5*	0/15
fminunc	**11**(16)	**25**(21)	∞	∞	∞	∞ *6e4*	0/15
simplex	63(71)	129(110)	∞	∞	∞	∞ *8e4*	0/15
OQNLP	15(19)	134(161)	∞	∞	∞	∞ *2e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f18	103	378	3968	9280	10905	12469	15/15
fmincon	16(17)	960(968)	∞	∞	∞	∞ *1e5*	0/15
fminunc	**11**(14)	412(461)	∞	∞	∞	∞ *7e4*	0/15
simplex	25(24)	754(827)	∞	∞	∞	∞ *8e4*	0/15
OQNLP	14(27)	**190**(208)	∞	∞	∞	∞ *2e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f19	1	1	242	1.2e5	1.2e5	1.2e5	15/15
fmincon	1(0)	**1**(0)	0.17(0.0)$_{\downarrow 4}$	∞	∞	∞ *1e5*	0/15
fminunc	**1**(0)	**1**(0)	0.13(0.0)$_{\downarrow 4}$	∞	∞	∞ *8e4*	0/15
simplex	**1**(0)	**1**(0)	0.18(0.0)$_{\downarrow 4}$	∞	∞	∞ *1e5*	0/15
OQNLP	1(0)	**1**(0)	**0.09**(0)$^{\star 4}_{\downarrow 4}$	**3.5**(4)	**3.6**(4)	∞ *3e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f20	16	851	38111	54470	54861	55313	14/15
fmincon	1.4(0)	6.4(9)	10(12)	7.3(8)	7.2(7)	7.2(8)	2/15
fminunc	**0.81**(0)$^{\star 4}$	**4.6**(4)	∞	∞	∞	∞ *6e4*	0/15
simplex	8.3(4)	5.4(3)	27(32)	19(23)	19(22)	19(21)	1/15
OQNLP	1.2(0)	5.8(7)	**3.5**(4)	**2.4**(3)	**2.4**(3)	**2.4**(3)	1/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f21	41	1157	1674	1705	1729	1757	14/15
fmincon	2.6(4)	**0.82**(1)	**0.94**(1)	**0.95**(1)	**0.97**(1)	**1.0**(1)	15/15
fminunc	2.3(5)	0.84(0.7)	0.98(1)	1.0(1)	1.1(1.0)	1.1(1.0)	15/15
simplex	16(15)	2.3(1)	3.6(4)	3.5(4)	3.5(4)	3.5(4)	15/15
OQNLP	**1.6**(2)	1.3(2)	1.8(2)	1.8(2)	1.8(2)	8.8(11)	2/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f22	71	386	938	1008	1040	1068	14/15
fmincon	**2.2**(3)	1.3(1)	**1.0**(1)	**1.00**(0.9)	**1.1**(0.8)	**1.2**(0.9)	15/15
fminunc	3.3(3)	**1.2**(1)	1.4(1)	1.4(1)	1.5(1)	1.5(1)	15/15
simplex	11(10)	5.1(3)	3.3(3)	3.1(3)	3.1(3)	3.0(3)	15/15
OQNLP	2.8(2)	2.5(3)	2.9(4)	2.8(4)	2.7(3)	9.4(10)	2/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f23	3.0	518	14249	31654	33030	34256	15/15
fmincon	**7.2**(6)	**1.6**(2)	**1.2**(2)	**10**(12)	∞	∞ *1e5*	0/15
fminunc	9.1(16)	10(10)	82(94)	∞	∞	∞ *8e4*	0/15
simplex	63(104)	3.0(1)	2.2(2)	23(24)	**22**(24)	**43**(49)	0/15
OQNLP	29(47)	5.6(4)	8.0(9)	∞	∞	∞ *4e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f24	1622	2.2e5	6.4e6	9.6e6	1.3e7	1.3e7	3/15
fmincon	3.0(3)	∞	∞	∞	∞	∞ *7e4*	0/15
fminunc	5.8(6)	∞	∞	∞	∞	∞ *7e4*	0/15
simplex	13(7)	5.9(6)	∞	∞	∞	∞ *9e4*	0/15
OQNLP	**0.73**(0.5)*	**0.17**(0.2)$^{\star 3}$	**0.02**(0.0)$^{\star 3}$	**0.01**(0.0)$^{\star 3}$	**0.01**(0.0)$^{\star 3}$	**0.01**(0.0)$^{\star 3}$	2/15

Table 1: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 (given in the respective first row) for different Δf values in dimension 5. The central 80% range divided by two is given in braces. The median number of conducted function evaluations is additionally given in *italics*, if $\mathrm{ERT}(10^{-7}) = \infty$. #succ is the number of trials that reached the final target $f_{\text{opt}} + 10^{-8}$. Best results are printed in bold.

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f1	43	43	43	43	43	43	15/15
fmincon	**0.77**(0.2)	1.7(0.5)	1.9(0.2)	2.8(0.2)	3.7(0.5)	4.7(0.5)	15/15
fminunc	1(0)	**1**(0)*	**1**(0)$^{\star 4}$	**1**(0)$^{\star 4}$	**1**(0)$^{\star 4}$	**1**(0)$^{\star 4}$	15/15
simplex	185(157)	6.4e4(7e4)	∞	∞	∞	∞ *4e5*	0/15
OQNLP	1.2(0)	1.3(0.2)	1.7(0.0)	1.8(0.0)	1.8(0)	1.8(0)	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f2	385	386	387	390	391	393	15/15
fmincon	5.6(3)	6.0(3)	**6.4**(2)	**7.4**(3)	**10**(3)	**14**(8)$^{\star 3}$	15/15
fminunc	19(4)	24(3)	37(28)	68(54)	80(78)	106(116)	15/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	**4.2**(2)	**5.3**(3)	6.5(3)	12(14)	13(14)	∞ *3e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f3	5066	7626	7635	7643	7646	7651	15/15
fmincon	∞	∞	∞	∞	∞	∞ *4e5*	0/15
fminunc	∞	∞	∞	∞	∞	∞ *4e5*	0/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	∞	∞	∞	∞	∞	∞ *3e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f4	4722	7628	7666	7700	7758	1.4e5	9/15
fmincon	∞	∞	∞	∞	∞	∞ *4e5*	0/15
fminunc	∞	∞	∞	∞	∞	∞ *4e5*	0/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	∞	∞	∞	∞	∞	∞ *3e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f5	41	41	41	41	41	41	15/15
fmincon	3.6(0)	5.2(0)	5.7(0)	6.7(0)	7.3(0)	103(110)	4/15
fminunc	2.1(0)	2.6(0)	2.6(0)	2.6(0)	2.6(0)	2.6(0)	15/15
simplex	172(94)	227(93)	227(93)	227(93)	227(93)	227(93)	15/15
OQNLP	**1.3**(0)$^{\star 4}$	**1.3**(0)$^{\star 4}$	**1.3**(0)$^{\star 4}$	**1.3**(0)$^{\star 4}$	**1.3**(0)$^{\star 4}$	**1.3**(0)$^{\star 4}$	15/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f6	1296	2343	3413	5220	6728	8409	15/15
fmincon	1.6(0.5)	1.4(0.5)	**1.4**(0.4)	**1.7**(0.4)$^{\star 3}$	**2.0**(0.4)$^{\star 3}$	**2.4**(0.4)$^{\star 4}$	15/15
fminunc	2.4(1)	2.7(0.7)	2.8(0.6)	3.1(0.6)	6.1(5)	724(777)	0/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	**1.0**(0.7)*	**1.4**(0.6)	1.5(0.7)	51(57)	∞	∞ *3e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f7	1351	4274	9503	16524	16524	16969	15/15
fmincon	∞	∞	∞	∞	∞	∞ *2e5*	0/15
fminunc	∞	∞	∞	∞	∞	∞ *2e5*	0/15
simplex	∞	∞	∞	∞	∞	∞ *3e5*	0/15
OQNLP	∞	∞	∞	∞	∞	∞ *2e4*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f8	2039	3871	4040	4219	4371	4484	15/15
fmincon	0.84(0.2)	0.84(0.8)	0.85(0.7)	0.85(0.7)	0.84(0.7)	**0.83**(0.7)*	15/15
fminunc	1.4(0.3)	1.1(0.1)	1.1(0.1)	1.1(0.1)	1.0(0.1)	1.0(0.1)	15/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	**0.67**(0.1)$_{\downarrow 3}$	**0.68**(0.4)$_{\downarrow}$	**0.70**(0.4)$_{\downarrow}$	**0.71**(0.3)$_{\downarrow}$	**0.70**(0.3)$_{\downarrow}$	∞ *3e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f9	1716	3102	3277	3455	3594	3727	15/15
fmincon	0.17(0.0)$_{\downarrow 4}$	0.34(0.0)$_{\downarrow 4}$	0.38(0.0)$_{\downarrow 4}$	0.40(0.0)$_{\downarrow 4}$	0.40(0.0)$_{\downarrow 4}$	0.40(0.0)$_{\downarrow 4}$	15/15
fminunc	**0.16**(0.0)$_{\downarrow 4}$	**0.26**(0.0)$_{\downarrow 4}$	**0.30**(0.0)$_{\downarrow 4}$	**0.33**(0.0)$_{\downarrow 4}$	**0.34**(0.0)$_{\downarrow 4}$	**0.34**(0.0)$_{\downarrow 4}$	15/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	0.32(1e-3)$_{\downarrow 4}$	0.27(1e-3)$_{\downarrow 4}$	0.31(1e-3)$_{\downarrow 4}$	0.34(4e-3)$_{\downarrow 4}$	0.34(4e-3)$_{\downarrow 4}$	5.1(7)	4/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f10	7413	8661	10735	14920	17073	17476	15/15
fmincon	**0.15**(0.0)$^{\star 3}_{\downarrow 4}$	**0.14**(0.0)$^{\star 3}_{\downarrow 4}$	**0.12**(0.0)$^{\star 4}_{\downarrow 4}$	**0.10**(0.0)$^{\star 4}_{\downarrow 4}$	34(48)	∞ *4e5*	0/15
fminunc	0.91(0.2)	1.1(0.7)	1.4(0.6)	2.4(2)	**10**(13)	∞ *4e5*	0/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	0.54(0.5)	0.98(0.9)	3.6(6)	61(67)	∞	∞ *3e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f11	1002	2228	6278	9762	12285	14831	15/15
fmincon	0.16(0.0)$_{\downarrow 4}$	**0.10**(0.0)$_{\downarrow 4}$	**0.04**(1e-2)$_{\downarrow 4}$	**1.1**(2)*	**228**(271)	∞ *4e5*	0/15
fminunc	0.51(0.1)$_{\downarrow 3}$	1.1(0.7)	0.97(0.7)	81(87)	∞	∞ *4e5*	0/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	**0.16**(0.0)$_{\downarrow 4}$	0.18(0.2)$_{\downarrow 4}$	0.25(0.2)$_{\downarrow 2}$	6.2(6)	∞	∞ *1e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f12	1042	1938	2740	4140	12407	13827	15/15
fmincon	0.81(0.5)	0.91(0.5)	0.84(0.5)	**0.85**(0.3)	**0.75**(0.8)	**14**(16)	4/15
fminunc	**0.61**(0.2)	**0.75**(0.5)	**0.80**(0.5)	0.94(0.4)	4.4(4)	61(75)	0/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	1.1(0.5)	2.5(0.7)	5.9(16)	24(35)	∞	∞ *3e5*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f13	652	2021	2751	18749	24455	30201	15/15
fmincon	1.1(0.2)	0.58(0.2)	0.79(0.6)	**0.55**(0.7)	∞	∞ *4e5*	0/15
fminunc	1.4(0.2)	0.96(0.1)	0.97(0.0)	20(32)	∞	∞ *4e5*	0/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	**1.1**(0.0)	**0.50**(0.0)$_{\downarrow 4}$	**0.48**(0.0)$_{\downarrow 4}$	22(23)	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f14	75	239	304	932	1648	15661	15/15
fmincon	**0.74**(0.3)	**0.49**(0.1)$^{\star 2}_{\downarrow 3}$	**0.65**(0.1)$_{\downarrow 2}$	0.65(0.1)$_{\downarrow 4}$	**0.68**(0.1)$^{\star 3}_{\downarrow 4}$	∞ *4e5*	0/15
fminunc	1.0(0.3)	0.98(0.4)	1.2(0.4)	0.86(0.1)$_{\downarrow}$	0.87(0.1)$_{\downarrow 2}$	∞ *4e5*	0/15
simplex	37(6)	8107(8363)	∞	∞	∞	∞ *4e5*	0/15
OQNLP	1.1(0.0)	0.62(0.1)$_{\downarrow 2}$	0.71(0.1)$_{\downarrow 2}$	**0.52**(0.1)$^{\star 3}_{\downarrow 4}$	∞	∞ *7e4*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f15	30378	1.5e5	3.1e5	3.2e5	4.5e5	4.6e5	15/15
fmincon	∞	∞	∞	∞	∞	∞ *4e5*	0/15
fminunc	∞	∞	∞	∞	∞	∞ *4e5*	0/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	∞	∞	∞	∞	∞	∞ *3e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f16	1384	27265	77015	1.9e5	2.0e5	2.2e5	15/15
fmincon	675(875)	∞	∞	∞	∞	∞ *4e5*	0/15
fminunc	∞	∞	∞	∞	∞	∞ *4e5*	0/15
simplex	**130**(121)	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	642(655)	∞	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f17	63	1030	4005	30677	56288	80472	15/15
fmincon	21(28)	∞	∞	∞	∞	∞ *4e5*	0/15
fminunc	**19**(38)	∞	∞	∞	∞	∞ *4e5*	0/15
simplex	25(15)	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	100(33)	∞	∞	∞	∞	∞ *3e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f18	621	3972	19561	67569	1.3e5	1.5e5	15/15
fmincon	∞	∞	∞	∞	∞	∞ *4e5*	0/15
fminunc	∞	∞	∞	∞	∞	∞ *4e5*	0/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	**1715**(1693)	∞	∞	∞	∞	∞ *3e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f19	1	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
fmincon	1(0)	**1**(0)	7.3e-4(0)$_{\downarrow 4}$	∞	∞	∞ *4e5*	0/15
fminunc	**1**(0)	**1**(0)	6.5e-4(3e-5)$_{\downarrow 4}$	∞	∞	∞ *4e5*	0/15
simplex	**1**(0)	**1**(0)	0.18(0.6)	∞	∞	∞ *4e5*	0/15
OQNLP	1(0)	**1**(0)	**5.4e-4**(1e-6)$^{\star 4}_{\downarrow 4}$	∞	∞	∞ *1e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f20	82	46150	3.1e6	5.5e6	5.6e6	5.6e6	14/15
fmincon	1.4(0)	10(9)	∞	∞	∞	∞ *4e5*	0/15
fminunc	**0.78**(0)$^{\star 4}$	**2.0**(2)$^{\star 2}$	∞	∞	∞	∞ *4e5*	0/15
simplex	88(62)	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	0.89(0)	23(25)	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f21	561	6541	14103	14643	15567	17589	15/15
fmincon	1.00(2)	**0.62**(0.8)	**0.46**(0.6)	**0.46**(0.6)	**0.46**(0.5)	**0.48**(0.5)*	9/15
fminunc	1.3(2)	0.82(1)	1.0(0.6)	1.0(0.6)	1.0(0.6)	10(13)	4/15
simplex	11(8)	90(85)	∞	∞	∞	∞ *4e5*	0/15
OQNLP	**0.55**(0.8)	1.8(4)	2.0(3)	1.9(2)	2.0(2)	∞ *5e4*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f22	467	5580	23491	24948	26847	1.3e5	12/15
fmincon	4.2(3)	**3.1**(3)	**4.8**(6)	**4.5**(6)	**4.3**(5)	**1.3**(1)	5/15
fminunc	**2.4**(3)	3.2(4)	7.0(9)	6.6(9)	6.2(8)	9.4(10)	1/15
simplex	64(91)	1073(1095)	∞	∞	∞	∞ *4e5*	0/15
OQNLP	3.1(4)	5.4(7)	8.3(11)	8.2(10)	27(32)	∞ *5e4*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f23	3.2	1614	67457	4.9e5	8.1e5	8.4e5	15/15
fmincon	**10**(13)	4.4(5)	**90**(94)	∞	∞	∞ *4e5*	0/15
fminunc	17(20)	248(254)	∞	∞	∞	∞ *4e5*	0/15
simplex	175(227)	**1.9**(0.2)	∞	∞	∞	∞ *4e5*	0/15
OQNLP	54(92)	3.5(4)	∞	∞	∞	∞ *2e5*	0/15
Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f24	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
fmincon	∞	∞	∞	∞	∞	∞ *4e5*	0/15
fminunc	∞	∞	∞	∞	∞	∞ *4e5*	0/15
simplex	∞	∞	∞	∞	∞	∞ *4e5*	0/15
OQNLP	∞	∞	∞	∞	∞	∞ *3e5*	0/15

Table 2: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 (given in the respective first row) for different Δf values in dimension 20. The central 80% range divided by two is given in braces. The median number of conducted function evaluations is additionally given in *italics*, if $\mathrm{ERT}(10^{-7}) = \infty$. #succ is the number of trials that reached the final target $f_{\text{opt}} + 10^{-8}$. Best results are printed in bold.