Multiobjectivization with NSGA-II on the Noiseless BBOB Testbed

Thanh-Do Tran[⋆•] Dimo Brockhoff[⋆] Bilel Derbel[⋆•]

[⋆]Inria Lille – Nord Europe, DOLPHIN team, 59650 Villeneuve d'Ascq, France
[•]Université Lille 1, LIFL, UMR CNRS 8022, 59655 Villeneuve d'Ascq Cedec, France
firstname.lastname@inria.fr

ABSTRACT

The idea of multiobjectivization is to reformulate a single-objective problem as a multiobjective one. In one of the scarce studies proposing this idea for problems in *continuous* domain, the distance to the closest neighbor (DCN) in the population of a multiobjective algorithm has been used as the additional (dynamic) second objective. As no comparison with other state-of-the-art single-objective optimizers for this idea has been presented, we benchmark two variants (with and without the second DCN objective) of the original NSGA-II algorithm using two different mutation operators on the noiseless BBOB'2013 testbed. It turns out that multiobjectivization helps for several of the 24 benchmark functions, but that, compared to the best algorithms from BBOB'2009, a significant performance loss is visible. Moreover, on some functions, the choice of mutation operator has a stronger impact on the performance than whether multiobjectivization is employed or not.

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, Multiobjectivization

1. INTRODUCTION

The idea of multiobjectivization, i.e., the reformulation of a single-objective problem by means of multiple objectives and its solution with a multiobjective optimizer, has been around since the beginning of the new millenium [14, 13]. Two basic ideas can thereby be distinguished: either the single-objective problem is *decomposed* into two or more objective functions [14] or one or more *additional* objective functions, so called *helper-objectives*, are optimized along with the original single-objective function [14, 13]. Several studies report on improving performance for combinatorial problems—early examples range from the traveling salesperson problem [14], over reducing bloat in genetic programming [3, 6], to job shop scheduling [13]. Also for some real world optimization tasks, multiobjectivization seems to help [9]. The main argument in favor of multiobjectivization for combinatorial problems is thereby the ability to overcome local optima and the possibility of introducing additional search directions on plateaus of equal function value.

Whereas the positive impact of multiobjectivization for combinatorial problems depends highly on the choice of typically problem-dependent objective functions [4, 10], for continuous problems, many studies favor a problem-independent approach, in which the diversity of the algorithm's population or archive is used as a second objective function [2, 5, 16, 17]. The main argument of [16] to use the distance to the closest neighbor (DCN) in the population of the NSGA-II algorithm [7] as the second objective is that such an objective function "decreases the selection pressure of the original [single-objective] optimisation scheme" with the result that "some low quality individuals could survive in the population with a higher probability" and in turn "these individuals could help to avoid stagnation in local optima" [16].

Unfortunately, in [16, 17] no comparison with other state-of-the-art single-objective methods have been performed. Here, we not only want to investigate the impact of multiobjectivization on the performance on the BBOB'2013 noiseless function testbed [8, 12], but also to see how the approach of [16] compares with the state-of-the-art algorithms for numerical optimization. To this end, we used the original implementation of NSGA-II [1] with the same algorithmic components as described in [16]. More precisely, we use no additional termination criterion other than the maximum number of function evaluations, perform no restarts, and use the proposed uniform mutation operator of [16]. In order to investigate the impact of multiobjectivization, we consider the NSGA-II variant with DCN as second objective (U-DCN) and the one where the second objective function is simply set to zero (U-zero). To have a better idea of how much the choice of the mutation operator affects the search performance, we further compare with the variants where NSGA-II's original polynomial mutation [7] is replaced with the uniform mutation (denoted by P-DCN and P-zero).

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

More details on the algorithms are given in the next section while Sec. 3 details the experimental procedure. Section 4 presents the mandatory timing experiments and the comparison results in Sec. 5 conclude the paper.

2. ALGORITHM PRESENTATION

2.1 The Artificial Second Objective

There are several ways to introduce artificial objectives into a mono-objective problem, which are in general to be considered as functions measuring the diversity of a population of solutions. In [17], the authors studied the performance of three such functions, namely DCN (distance to the closest neighbor of the population), ADI (average distance de all individuals), and DBI (distance to the best individual). They showed that multiobjectivization with DCN as a second objective leads to superior performance.

In this study, we shall also use DCN for multiobjectivizing the BBOB functions. Having a set of individuals, the DCN with respect to individual i is defined as the Euclidean distance to the closest member of the population, considering that the decision space is real-valued. More formally,

$$\text{DCN}(i) = \min_{j \neq i} \left\{ \Big(\sum_{\ell} (x^i_\ell - x^j_\ell)^2 \Big)^{\frac{1}{2}} \right\}$$

where x^i_ℓ is ℓ^{th} decision variable with respect to individual i.

2.2 The Multiobjective Algorithm

Among a multitude of multiobjective algorithms, we consider the well known NSGA-II [7] which has also been used in [17]. For the sake of reproducibility, we recall the main components of NSGA-II and the way their are implemented in our experiments. First, since the standard NSGA-II deals with minimizing objectives and DCN is to be maximized (i.e., to increase diversity), we set the second objective used in our NSGA-II to be the maximum DCN over all individuals minus the DCN of the considered individual. The population size N is set to be 8. In fact, a small population size was shown to perform relatively good in [17], and the population size in the standard NSGA-II implementation [1] should be a multiple of 4. We use the Simulated Binary Crossover (SBX) with a distribution index of 15, and where each gene (variable) is crossed with a probability of 0.5. As for the mutation, we consider two operators: (i) the Uniform Mutation (U), and the Polynomial Mutation (P) with distribution index $\eta = 100$. Notice that only the Uniform Mutation has been considered in [17]. We set the crossover probability to 1 and the mutation probability to $1/D$, where D is the number of variables (i.e. problem dimensions). While the uniform mutation naturally restricts the variables to an interval (here chosen as $[-5, 5]$), we also restrict the decision variables to this interval for the polynomial mutation by assigning all the mass of the probability distribution that is outside a variable bound to the boundary value.

3. EXPERIMENTAL PROCEDURE

To study the impact of multiobjectivization, we consider running NSGA-II while artificially setting the second objective to zero, i.e., all individuals have equal values in their second objectives. This has the effect of turning off the crowding distance based selection mechanism specific to NSGA-II, and favoring the selection of individuals having better fitness in the original first objective.

We end up with four algorithm variants depending on whether DCN is switched on or off, and which mutation (U or P) is used. In the remainder, these variants are respectively denoted by U-DCN, U-zero, P-DCN and P-zero.

For our experimentations, we use the standard C implementation of NSGA-II available for free download at [1], and setting objectives and parameters to fit in our settings. Moreover, the initial population in NSGA-II is uniformly sampled in $[-5, 5]^D$. We run the four variants NSGA-II up to a budget of $10^6 D$ function evaluations or until the maximal BBOB precision of 10^{-8} is reached. It is to notice that there is no independent restart in our algorithms. We considered dimensions $D \in \{2, 3, 5, 10, 20\}$ and all the 15 instances of BBOB'2013.

4. TIMING EXPERIMENTS

In order to assess the dependency of the four algorithm variants on the problem dimension, the requested BBOB timing experiments were performed on a Dell XPS 720 machine using the Intel® Core™2 Quad Processor Q6600 running at 2.40 GHz with 2.0 GiB RAM. Note that each implementation was deployed exclusively on a single core of the CPU. All implementations were built using the GCC 4.7.2 compiler and executed under the Ubuntu 12.10 Linux distribution. Each algorithm variant was run iteratively on the first instance of f_8 within $10^5 D$ function evaluations until *at least* 30 seconds had passed. This procedure was repeated over seven problem dimensions, i.e. $D = \{2, 3, 5, 10, 20, 40, 80\}$. The approximate per-function-evaluation runtimes for U-zero were 7.3, 7.8, 8.5, 10, 13, 18, and 29 times 10^{-7} seconds; for U-DCN12, 13, 16, 23, 38, 90, and 110 times 10^{-7} seconds; for P-zero 9.9, 10, 11, 13, 16, 23, and 36 times 10^{-7} seconds; and for P-DCN13, 16, 19, 28, 42, 58, and 82 times 10^{-7} seconds in 2, 3, 5, 10, 20, 40, and 80 dimensions respectively.

5. RESULTS

Results from experiments according to [11] on the benchmark functions given in [8, 12] 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 [11, 15]. **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.

From our experimental results, three main observations can be formulated.

Impact of Multiobjectivization: When comparing the DCN variants to the variants without DCN, multiobjectivization seems to help on some functions whereas a negative impact can only be observed for a few. For NSGA-II with uniform mutation, U-DCN outperforms U-zero for the separable functions, original Rosenbrock (f_8) and the discus function (f_{11} in 20D) for almost all targets and on f_{14} and f_{22} in 5D and for the most difficult targets. U-zero is, on the other hand, only better on f_{14} for easy targets. For NSGA-II with polynomial mutation, the impact of multiobjectivization is less pronounced with a similar tendency on the separable and moderate functions with P-DCN being better only on f_2, f_6, f_8, and f_{14}. On the sphere (f_1) and the linear function f_5, on the other hand, the version without the DCN objective is clearly better. The performance differences are larger in higher dimensions. All other results are statistically not significant[1].

Impact of Mutation: On some of the functions, the choice of the mutation operator seems to have a stronger impact on the performance than whether multiobjectivization is employed or not. When comparing U-DCN with P-DCN, the polynomial mutation is giving better results on functions f_1, f_2, f_5, f_6, f_{11}, and f_{14} for the most difficult targets while the uniform mutation is typically better at the beginning of the search. This is not surprising as the distribution index $\eta = 100$ for the polynomial mutation is fixed throughout the search, which means that the mutation's step size is typically too small at the beginning of the search but better suited at the later stages. On f_{20}, f_{21}, and f_{22}, the uniform mutation is, however, interestingly better for all targets.

Competitiveness: The third observation is that the ERTs of all the four variants are still far from being competitive with the artificial best algorithm from BBOB'2009.

It is worth noticing that, initially, our goal was *not* to design an optimizer that would perform competitively compared to existing state-of-the-art single-objective algorithms, which normally use advanced optimization techniques like step size adaptation. Notice also that we have conducted some other preliminary experiments using another advanced multiobjective algorithm, namely R2-EMOA, and we observed better ERTs, but a seemingly comparable impact of DCN. This suggests that DCN or any other alternative objectives may, to some extent, be beneficial if carefully combined with an appropriate multiobjective algorithm or variation operator (e.g. a specific mutation). From our experiments, however, we can only conclude that using DCN with the specified NSGA-II is showing a limited potential for tackling the noiseless single-objective BBOB testbed.

Acknowledgements

The authors acknowledge support by the French national research agency (ANR) within the Modèles Numérique project "NumBBO - Analysis, Improvement and Evaluation of Numerical Blackbox Optimizers" (ANR-12-MONU-0009-03).

6. REFERENCES

[1] NSGA-II C code implementation. `http://www.iitk.ac.in/kangal/codes.shtml`.

[2] H. A. Abbass and K. Deb. Searching under Multi-evolutionary Pressures. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, p. 391–404, 2003.

[3] S. Bleuler, M. Brack, L. Thiele, and E. Zitzler. Multiobjective Genetic Programming: Reducing Bloat by Using SPEA2. In *Congress on Evolutionary Computation (CEC 2001)*, p. 536–543, 2001.

[4] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. On the Effects of Adding Objectives to Plateau Functions. *IEEE Transactions on Evolutionary Computation*, 13(3):591–603, 2009.

[5] L. T. Bui, H. A. Abbass, and J. Branke. Multiobjective Optimization for Dynamic Environments. In *Congress on Evolutionary Computation (CEC 2005)*, p. 2349–2356, 2005.

[6] E. D. de Jong, R. A. Watson, and J. B. Pollack. Reducing Bloat and Promoting Diversity using Multi-Objective Methods. In *Genetic and Evolutionary Computation Conference (GECCO 2001)*, p. 11–18. 2001.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[8] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Tech. Report 2009/20, Research Center PPE, 2009. Updated February 2010.

[9] J. Handl, S. C. Lovell, and J. Knowles. Investigations into the Effect of Multiobjectivization in Protein Structure Prediction. In *Conference on Parallel Problem Solving From Nature (PPSN X)*, p. 702–711. 2008.

[10] J. Handl, S. C. Lovell, and J. Knowles. Multiobjectivization by Decomposition of Scalar Cost Functions. In *Conference on Parallel Problem Solving From Nature (PPSN X)*, p. 31–40. 2008.

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

[12] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Tech. Report RR-6829, INRIA, 2009. Updated February 2010.

[13] M. T. Jensen. Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation. *J. of Math. Modelling and Algorithms*, 3(4):323–347, 2004.

[14] J. D. Knowles, R. A. Watson, and D. W. Corne. Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, p. 269–283, 2001.

[15] K. Price. Differential evolution vs. the functions of the second ICEO. In *Congress on Evolutionary Computation (CEC 1997)*, p. 153–157, 1997.

[16] E. Segredo, C. Segura, and León. Analysing the Robustness of Multiobjectivisation Parameters with Large Scale Optimisation Problems. In *Congress on Evolutionary Computation (CEC 2012)*, p. 1–8, 2012.

[17] C. Segura, E. Segredo, and C. León. Analysing the robustness of multiobjectivisation approaches applied to large scale optimisation problems. In *EVOLVE- A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation*, p. 365–391. 2013.

[1]Note that the significance levels were checked by comparing two algorithms at a time (not shown here).

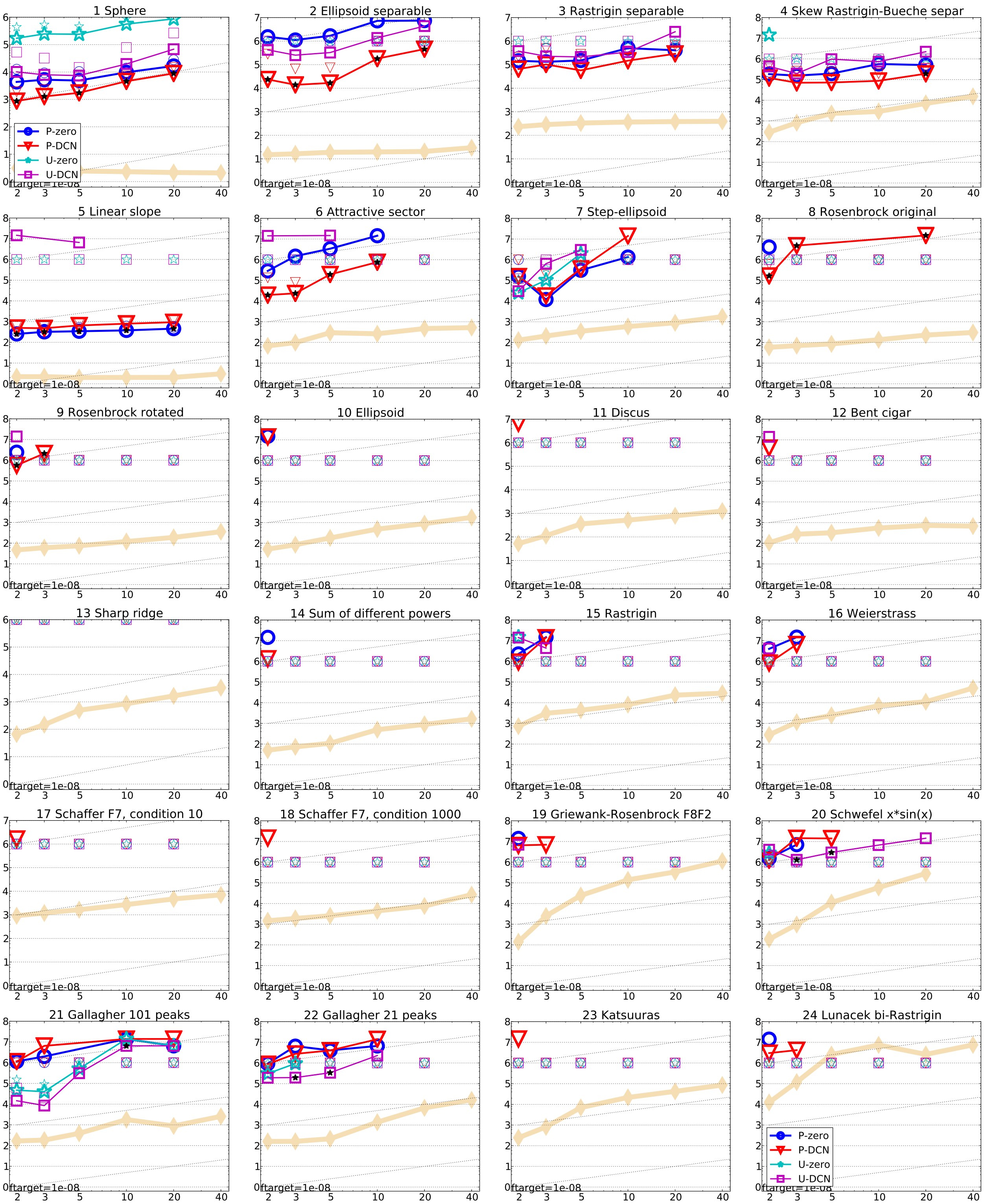

Figure 1: Expected running time (ERT in number of f-evaluations) divided by dimension for target function value 10^{-8} as $\log_{10}$ values versus dimension. Different symbols correspond to different algorithms given in the legend of f_1 and f_{24}. Light symbols give the maximum number of function 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: ○:P-zero, ▽:P-DCN, ⋆:U-zero, □:U-DCN

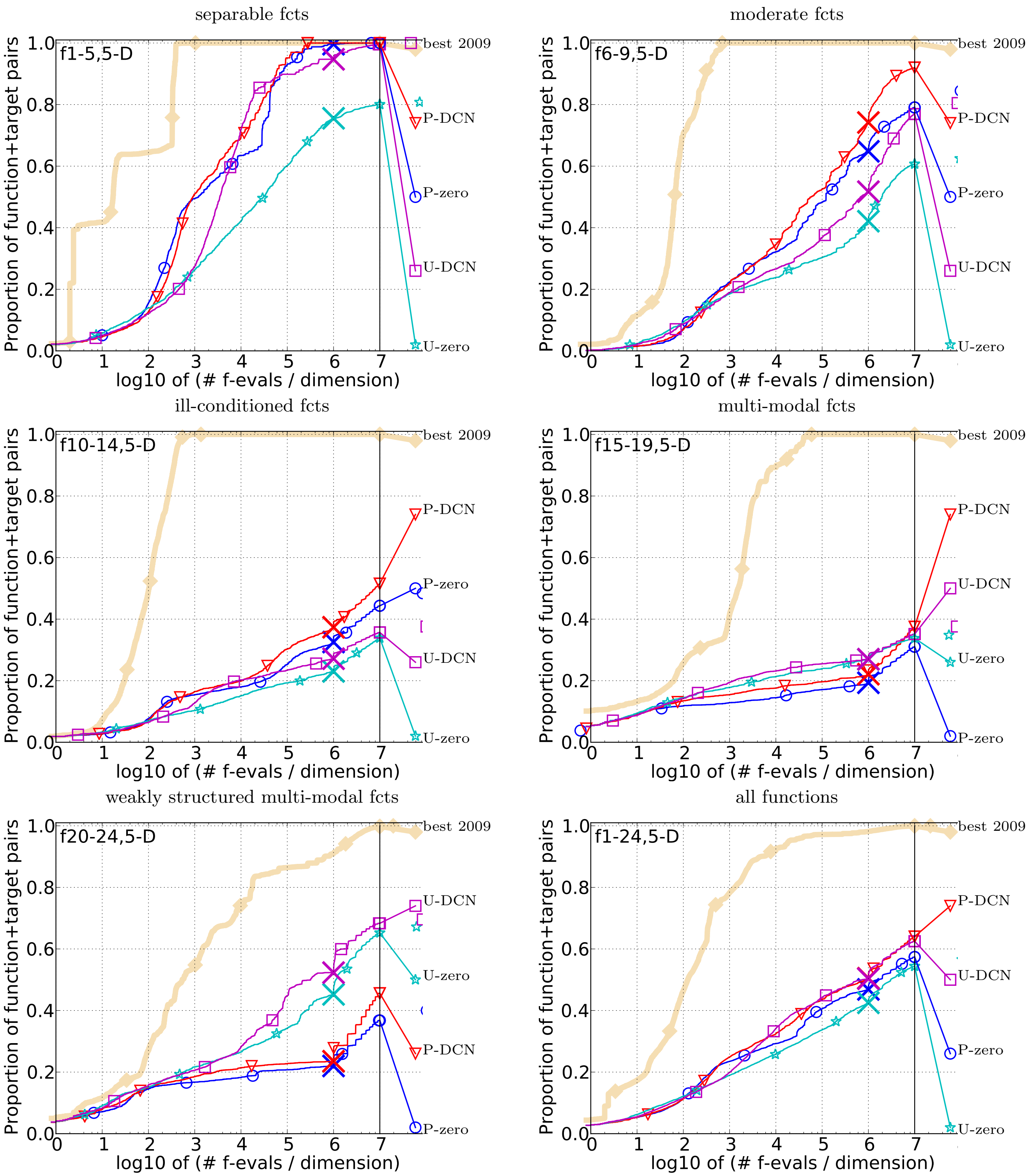

Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 5-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single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

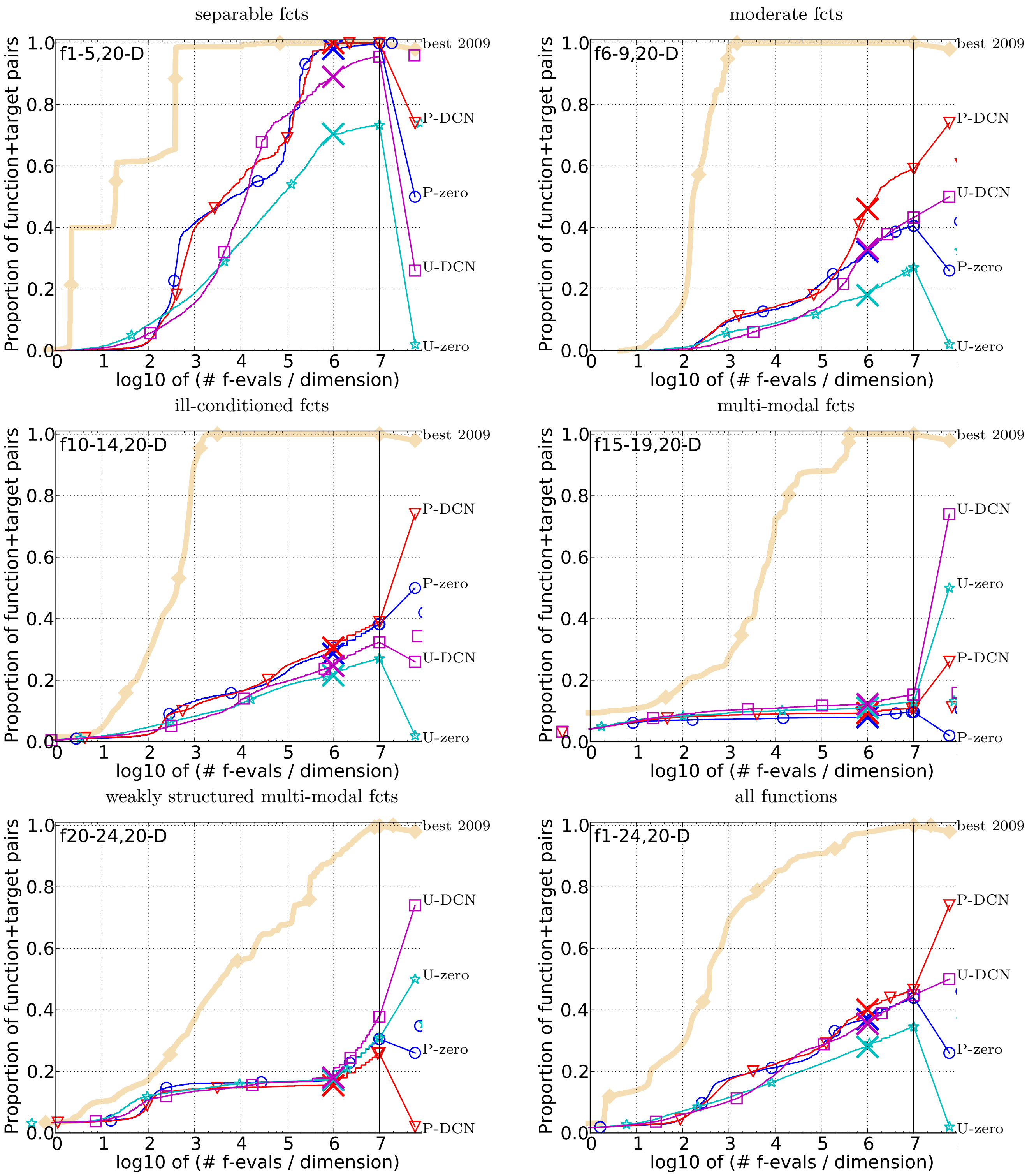

Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 20-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single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

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f1	11	12	12	12	12	12	15/15
P-0	11(14)	29(26)	**41**(30)	**55**(27)	**133**(35)$^{\star 2}$	664(236)	15/15
P-D	7.5(12)	28(20)	41(20)	64(25)	199(48)	**494**(147)	15/15
U-0	3.2(3)	**14**(8)	54(43)	381(303)	4039(2899)	3.9e4(2e4)	15/15
U-D	**3.1**(3)	21(11)	84(45)	367(148)	848(548)	2283(1702)	15/15
f2	83	87	88	90	92	94	15/15
P-0	**16**(10)	**21**(11)	**44**(32)	225(135)	2835(3668)	1.8e4(2e4)	7/15
P-D	20(9)	27(9)	47(22)	**139**(99)	**212**(111)$^{\star 3}$	**610**(818)*	15/15
U-0	175(246)	780(964)	1961(1617)	2.0e4(2e4)	∞	∞ *5e6*	0/15
U-D	107(160)	193(191)	264(225)	1229(858)	1345(776)	7404(6761)	12/15
f3	716	1622	1637	1646	1650	1654	15/15
P-0	9.4(9)	44(27)	111(41)	111(41)	116(41)	223(75)	15/15
P-D	7.5(11)	26(29)	152(178)	152(177)	155(176)	**166**(173)	15/15
U-0	**0.62**(0.5)	**2.4**(1)	8.4(5)	88(61)	1027(650)	∞ *5e6*	0/15
U-D	0.85(0.5)	3.6(2)	**5.4**(3)	**10**(4)$^{\star 2}$	**65**(62)	374(678)	13/15
f4	809	1633	1688	1817	1886	1903	15/15
P-0	24(9)	81(40)	139(52)	131(49)	132(58)	256(95)	15/15
P-D	42(46)	74(116)	210(286)	195(266)	189(258)	**189**(255)	15/15
U-0	**0.76**(0.4)	**4.7**(2)	12(10)	112(97)	1134(559)	∞ *5e6*	0/15
U-D	1.1(0.7)	6.1(4)	**8.2**(6)	**12**(6)$^{\star 2}$	**89**(192)	876(1352)	8/15
f5	10	10	10	10	10	10	15/15
P-0	50(58)	**101**(85)	**113**(84)	**129**(84)	**146**(81)	**162**(81)$^{\star 3}$	15/15
P-D	37(29)	118(77)	132(74)	168(69)	219(50)	290(56)	15/15
U-0	**19**(15)	217(97)	2697(1161)	2.5e5(1e5)	∞	∞ *5e6*	0/15
U-D	25(18)	240(136)	988(587)	3.7e4(8e4)	2.6e5(4e5)	1.5e6(2e6)	2/15
f6	114	214	281	580	1038	1332	15/15
P-0	7.3(7)	**6.4**(5)	8.8(10)	109(104)	1887(2491)	6888(8649)	4/15
P-D	7.6(7)	7.6(4)	**8.3**(5)	**23**(16)	**75**(88)	**410**(303)$^{\star 3}$	14/15
U-0	**6.5**(7)	77(174)	1678(1399)	8123(9346)	9430(1e4)	∞ *5e6*	0/15
U-D	7.7(9)	96(189)	293(492)	2097(4324)	3875(5032)	5.3e4(6e4)	1/15
f7	24	324	1171	1572	1572	1597	15/15
P-0	217(261)	2798(7738)	**1284**(2159)	**983**(1602)	**983**(1646)	**968**(1595)	13/15
P-D	113(233)	1695(2334)	1569(2303)	1175(1774)	1175(1813)	1158(1747)	12/15
U-0	**12**(13)	1488(2964)	2969(3523)	6279(6881)	6279(7379)	6188(6416)	6/15
U-D	20(22)	**1132**(41)	1421(2326)	9508(1e4)	9508(1e4)	9362(1e4)	4/15
f8	73	273	336	391	410	422	15/15
P-0	18(19)	2863(9152)	**3230**(7738)	1.8e5(2e5)	1.7e5(2e5)	∞ *5e6*	0/15
P-D	23(24)	6797(9224)	6645(7523)	**9250**(6704)	**1.4e4**(6794)	∞ *5e6*	0/15
U-0	**16**(9)	7820(1e4)	∞	∞	∞	∞ *5e6*	0/15
U-D	17(16)	**713**(1962)	4007(4023)	1.9e5(2e5)	∞	∞ *5e6*	0/15
f9	35	127	214	300	335	369	15/15
P-0	685(1900)	**3145**(607)	**3476**(1371)	2.4e5(3e5)	∞	∞ *5e6*	0/15
P-D	446(980)	2.7e4(4e4)	1.7e4(2e4)	**1.9e4**(2e4)	**5.1e4**(5e4)	**2.0e5**(2e5)	0/15
U-0	378(55)	1.4e4(2e4)	3.4e5(4e5)	∞	∞	∞ *5e6*	0/15
U-D	**65**(59)	3.1e4(4e4)	2.3e4(4e4)	7.2e4(8e4)	∞	∞ *5e6*	0/15
f10	349	500	574	626	829	880	15/15
P-0	2.3e4(2e4)	1.4e5(2e5)	∞	∞	∞	∞ *5e6*	0/15
P-D	**7421**(8952)	**6.6e4**(8e4)	**1.3e5**(1e5)	**1.2e5**(1e5)	∞	∞ *5e6*	0/15
U-0	2.1e5(2e5)	∞	∞	∞	∞	∞ *5e6*	0/15
U-D	9.9e4(1e5)	∞	∞	∞	∞	∞ *5e6*	0/15
f11	143	202	763	1177	1467	1673	15/15
P-0	781(856)	1264(814)	479(270)	797(227)	5.1e4(5e4)	∞ *5e6*	0/15
P-D	606(502)	**896**(629)	**312**(165)	**619**(310)	**1.2e4**(1e4)	∞ *5e6*	0/15
U-0	447(538)	2846(3800)	9233(8613)	∞	∞	∞ *5e6*	0/15
U-D	**239**(214)	2586(3033)	8587(9687)	∞	∞	∞ *5e6*	0/15
f12	108	268	371	461	1303	1494	15/15
P-0	9.2e4(1e5)	1.2e5(1e5)	8.8e4(1e5)	∞	∞	∞ *5e6*	0/15
P-D	**1.2e4**(2e4)	**2.8e4**(4e4)	1.9e5(2e5)	∞	∞	∞ *5e6*	0/15
U-0	2.4e4(5e4)	3.8e4(5e4)	8.9e4(1e5)	∞	∞	∞ *5e6*	0/15
U-D	1.7e4(2e4)	3.7e4(5e4)	**8.8e4**(1e5)	∞	∞	∞ *5e6*	0/15
f13	132	195	250	1310	1752	2255	15/15
P-0	7.6e4(9e4)	1.7e5(2e5)	2.8e5(3e5)	∞	∞	∞ *5e6*	0/15
P-D	3.4e4(4e4)	3.6e5(4e5)	**2.8e5**(3e5)	∞	∞	∞ *5e6*	0/15
U-0	2.5e4(4e4)	1.7e5(2e5)	2.8e5(3e5)	∞	∞	∞ *5e6*	0/15
U-D	**1.4e4**(2e4)	**5.1e4**(6e4)	2.8e5(3e5)	∞	∞	∞ *5e6*	0/15
f14	10	41	58	139	251	476	15/15
P-0	3.1(4)	11(7)	11(5)	**80**(54)	∞	∞ *5e6*	0/15
P-D	2.0(2)	9.3(4)	**10**(3)	113(191)	**5.4e4**(5e4)$^{\star 2}$	∞ *5e6*	0/15
U-0	1.5(2)	**3.6**(2)	11(6)	2.6e5(3e5)	∞	∞ *5e6*	0/15
U-D	**1.2**(0.7)	6.8(5)	23(12)	1269(1067)	∞	∞ *5e6*	0/15
f15	511	9310	19369	20073	20769	21359	14/15
P-0	4.5e4(5e4)	∞	∞	∞	∞	∞ *5e6*	0/15
P-D	6.4e4(8e4)	∞	∞	∞	∞	∞ *5e6*	0/15
U-0	1553(2758)	∞	∞	∞	∞	∞ *5e6*	0/15
U-D	**516**(62)	**2163**(2685)	∞	∞	∞	∞ *5e6*	0/15
f16	120	612	2662	10449	11644	12095	15/15
P-0	5.5(16)	7997(1e4)	1.2e4(2e4)	∞	∞	∞ *5e6*	0/15
P-D	**1.1**(0.7)	1846(4086)	7522(8462)	**3290**(3652)	∞	∞ *5e6*	0/15
U-0	1.3(0.9)	600(1235)	2030(2812)	∞	∞	∞ *5e6*	0/15
U-D	1.7(1.0)	**90**(82)	**1913**(2389)	∞	∞	∞ *5e6*	0/15
f17	5.2	215	899	3669	6351	7934	15/15
P-0	5.6(8)	3.3e4(4e4)	2.3e4(3e4)	∞	∞	∞ *5e6*	0/15
P-D	4.3(4)	4.7e4(6e4)	7.8e4(9e4)	∞	∞	∞ *5e6*	0/15
U-0	**2.6**(2)	33(56)	**4877**(6051)	∞	∞	∞ *5e6*	0/15
U-D	4.1(4)	**6.0**(9)	5087(8339)	∞	∞	∞ *5e6*	0/15
f18	103	378	3968	9280	10905	12469	15/15
P-0	933(2547)	4.0e4(5e4)	1.8e4(2e4)	∞	∞	∞ *5e6*	0/15
P-D	193(272)	2.9e4(3e4)	8439(9400)	∞	∞	∞ *5e6*	0/15
U-0	**2.1**(2)	7032(1e4)	8805(1e4)	∞	∞	∞ *5e6*	0/15
U-D	2.5(3)	**6288**(7517)	**8405**(1e4)	∞	∞	∞ *5e6*	0/15
f19	1	1	242	1.2e5	1.2e5	1.2e5	15/15
P-0	**27**(16)	2.2e6(3e6)	3.1e5(3e5)	∞	∞	∞ *5e6*	0/15
P-D	44(40)	5.0e5(9e5)	6.5e4(7e4)	**619**(706)	**618**(640)	∞ *5e6*	0/15
U-0	**27**(20)	**7251**(6858)	**3.2e4**(4e4)	∞	∞	∞ *5e6*	0/15
U-D	32(30)	1.3e4(1e4)	4.9e4(5e4)	∞	∞	∞ *5e6*	0/15
f20	16	851	38111	54470	54861	55313	14/15
P-0	15(10)	371(909)	∞	∞	∞	∞ *5e6*	0/15
P-D	8.6(8)	336(965)	1837(2198)	1285(1423)	1276(1504)	1266(1559)	1/15
U-0	**4.0**(2)	17(29)	527(655)	369(413)	371(413)	390(411)	0/15
U-D	4.5(3)	**5.6**(8)	**367**(459)	**257**(321)	**255**(276)	**261**(332)	4/15
f21	41	1157	1674	1705	1729	1757	14/15
P-0	8713(3)	1.2e4(2e4)	∞	∞	∞	∞ *5e6*	0/15
P-D	1.9e4(6e4)	1.2e4(2e4)	∞	∞	∞	∞ *5e6*	0/15
U-0	2.9(1)	**236**(396)	1409(2021)	1386(1897)	1376(1871)	1398(1834)	11/15
U-D	**2.7**(3)	435(447)	**906**(1582)	**892**(1553)	**885**(1532)	**880**(1503)	12/15
f22	71	386	938	1008	1040	1068	14/15
P-0	2.6e4(4e4)	3.6e4(5e4)	2.1e4(3e4)	2.0e4(3e4)	1.9e4(2e4)	1.9e4(3e4)	3/15
P-D	2.6e4(4e4)	1.9e4(3e4)	2.1e4(2e4)	2.0e4(2e4)	1.9e4(2e4)	1.9e4(2e4)	3/15
U-0	4.8(6)	1106(458)	1458(2755)	1832(2640)	5409(5342)	3.5e4(4e4)	0/15
U-D	**1.9**(1)	**327**(164)	**1146**(2667)	**1129**(2513)	**1256**(2478)	**1469**(2343)	13/15
f23	3.0	518	14249	31654	33030	34256	15/15
P-0	3.6(3)	29(62)	2398(2711)	∞	∞	∞ *5e6*	0/15
P-D	3.6(3)	**3.2**(4)	**970**(1228)	∞	∞	∞ *5e6*	0/15
U-0	**3.2**(2)	43(50)	1527(1678)	∞	∞	∞ *5e6*	0/15
U-D	3.4(2)	10(17)	∞	∞	∞	∞ *5e6*	0/15
f24	1622	2.2e5	6.4e6	9.6e6	1.3e7	1.3e7	3/15
P-0	∞	∞	∞	∞	∞	∞ *5e6*	0/15
P-D	1155(1651)	∞	∞	∞	∞	∞ *5e6*	0/15
U-0	**575**(655)	∞	∞	∞	∞	∞ *5e6*	0/15
U-D	709(989)	∞	∞	∞	∞	∞ *5e6*	0/15

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 $f_{\mathrm{opt}} + 10^{-8}$. Best results are printed in bold. Algorithm names are abbreviated (e.g. P-0 is P-zero).

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f1	43	43	43	43	43	43	15/15
P-0	45(15)	63(15)	**71**(12)	**100**(22)$^{\star 3}$	**337**(79)$^{\star 4}$	**2140**(550)	15/15
P-D	44(14)	63(16)	74(17)	147(38)	702(169)	2421(596)	15/15
U-0	**8.6**(2)$^{\star 4}$	**37**(10)$^{\star 3}$	119(26)	1029(335)	1.0e4(3953)	1.1e5(3e4)	13/15
U-D	21(5)	111(30)	378(97)	1589(413)	4632(1517)	1.0e4(2945)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f2	385	386	387	390	391	393	15/15
P-0	**22**(3)	**28**(10)	69(64)	637(212)	4598(2229)	1.6e5(2e5)	2/15
P-D	25(6)	32(6)	**52**(14)	**257**(164)$^{\star 3}$	**1017**(1191)$^{\star 3}$	**3815**(4266)$^{\star 2}$	13/15
U-0	305(203)	1104(917)	4015(1999)	2.8e4(1e4)	∞	∞ *2e7*	0/15
U-D	130(48)	374(352)	563(333)	1535(1733)	2.1e4(3e4)	9.5e4(1e5)	3/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f3	5066	7626	7635	7643	7646	7651	15/15
P-0	216(157)	283(156)	378(171)	378(171)	381(169)	**508**(147)	15/15
P-D	145(75)	541(590)	753(690)	752(689)	752(689)	751(688)	15/15
U-0	**1.4**(0.3)$^{\star 3}$	**4.8**(2)$^{\star 2}$	15(6)	170(81)	1900(536)	∞ *2e7*	0/15
U-D	2.7(1)	9.5(5)	**13**(4)	**43**(26)$^{\star 4}$	**138**(117)$^{\star 2}$	2113(2614)	5/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f4	4722	7628	7666	7700	7758	1.4e5	9/15
P-0	271(152)	263(100)	375(134)	373(134)	374(130)	28(7)	15/15
P-D	214(249)	337(323)	447(313)	445(312)	444(309)	**25**(17)	15/15
U-0	**2.1**(0.6)$^{\star 2}$	**6.9**(2)$^{\star 2}$	25(6)	255(69)	3201(1837)	∞ *2e7*	0/15
U-D	3.9(2)	12(8)	**16**(6)$^{\star 2}$	**42**(15)$^{\star 3}$	**219**(173)	120(119)	5/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f5	41	41	41	41	41	41	15/15
P-0	128(20)	**163**(32)	**174**(31)	**185**(30)$^{\star 3}$	**200**(27)$^{\star 4}$	**217**(29)$^{\star 4}$	15/15
P-D	141(31)	190(46)	204(47)	248(29)	336(37)	423(35)	15/15
U-0	**92**(24)$^{\star 2}$	858(334)	1.1e4(2402)	∞	∞	∞ *2e7*	0/15
U-D	174(77)	942(290)	3043(1126)	1.6e4(5662)	1.3e6(1e6)	∞ *2e7*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f6	1296	2343	3413	5220	6728	8409	15/15
P-0	**9.2**(4)	826(2658)	3139(4113)	1.6e4(2e4)	∞	∞ *2e7*	0/15
P-D	16(3)	**17**(35)	**471**(100)	**2005**(2293)	**4356**(4185)	**1.1e4**(1e4)	0/15
U-0	3013(7731)	7233(1e4)	8.2e4(1e5)	5.5e4(6e4)	∞	∞ *2e7*	0/15
U-D	358(384)	2573(3520)	3426(3787)	6585(6094)	∞	∞ *2e7*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f7	1351	4274	9503	16524	16524	16969	15/15
P-0	**2528**(2942)	**1.7e4**(2e4)$^{\star 3}$	∞	∞	∞	∞ *2e7*	0/15
P-D	1.7e4(2e4)	∞	∞	∞	∞	∞ *2e7*	0/15
U-0	2.1e4(2e4)	∞	∞	∞	∞	∞ *2e7*	0/15
U-D	1.8e4(2e4)	∞	∞	∞	∞	∞ *2e7*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f8	2039	3871	4040	4219	4371	4484	15/15
P-0	**215**(128)	**653**(65)	**959**(81)	∞	∞	∞ *2e7*	0/15
P-D	563(316)	1531(2658)	1717(2556)	**2408**(2397)	**3376**(2378)*	**1.3e4**(1e4)*	1/15
U-0	1382(1246)	2430(2967)	2.0e4(2e4)	∞	∞	∞ *2e7*	0/15
U-D	551(579)	767(712)	1198(748)	2494(967)	3.3e4(4e4)	∞ *2e7*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f9	1716	3102	3277	3455	3594	3727	15/15
P-0	**412**(296)$^{\star 2}$	**4135**(3363)	**6472**(6053)	∞	∞	∞ *2e7*	0/15
P-D	1600(1101)	5404(3702)	∞	∞	∞	∞ *2e7*	0/15
U-0	4.0e4(4e4)	∞	∞	∞	∞	∞ *2e7*	0/15
U-D	1.7e4(1e4)	9.3e4(1e5)	∞	∞	∞	∞ *2e7*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f10	7413	8661	10735	14920	17073	17476	15/15
P-0	∞	∞	∞	∞	∞	∞ *2e7*	0/15
P-D	∞	∞	∞	∞	∞	∞ *2e7*	0/15
U-0	∞	∞	∞	∞	∞	∞ *2e7*	0/15
U-D	∞	∞	∞	∞	∞	∞ *2e7*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f11	1002	2228	6278	9762	12285	14831	15/15
P-0	676(222)	617(254)	349(109)	1233(468)	∞	∞ *2e7*	0/15
P-D	**457**(265)	**417**(172)	**211**(78)$^{\star 2}$	**850**(385)	∞	∞ *2e7*	0/15
U-0	1213(560)	7556(4994)	∞	∞	∞	∞ *2e7*	0/15
U-D	786(641)	3035(2507)	6177(5055)	∞	∞	∞ *2e7*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f12	1042	1938	2740	4140	12407	13827	15/15
P-0	6998(9606)	1.6e4(2e4)	**4.7e4**(6e4)	∞	∞	∞ *2e7*	0/15
P-D	7018(9612)	4.1e4(5e4)	1.0e5(1e5)	**6.8e4**(7e4)	∞	∞ *2e7*	0/15
U-0	7490(9691)	**1.1e4**(2e4)	1.0e5(1e5)	∞	∞	∞ *2e7*	0/15
U-D	**3164**(9601)	1.2e4(2e4)	4.8e4(6e4)	∞	∞	∞ *2e7*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f13	652	2021	2751	18749	24455	30201	15/15
P-0	2.7e4(5e4)	4.0e4(5e4)	**4.7e4**(6e4)	**1.6e4**(2e4)	∞	∞ *2e7*	0/15
P-D	1.5e4(3e4)	**1.5e4**(2e4)	1.0e5(1e5)	∞	∞	∞ *2e7*	0/15
U-0	**5052**(2e4)	1.6e4(2e4)	1.0e5(1e5)	∞	∞	∞ *2e7*	0/15
U-D	2.7e4(3e4)	4.0e4(5e4)	1.0e5(1e5)	∞	∞	∞ *2e7*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f14	75	239	304	932	1648	15661	15/15
P-0	25(9)	15(4)	14(3)	3203(2945)	∞	∞ *2e7*	0/15
P-D	22(8)	15(3)	**14**(2)	**1149**(358)*	∞	∞ *2e7*	0/15
U-0	**4.0**(2)*	**6.2**(2)$^{\star 4}$	22(7)	∞	∞	∞ *2e7*	0/15
U-D	6.6(3)	15(3)	54(15)	2814(1041)	∞	∞ *2e7*	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
P-0	∞	∞	∞	∞	∞	∞ *2e7*	0/15
P-D	∞	∞	∞	∞	∞	∞ *2e7*	0/15
U-0	∞	∞	∞	∞	∞	∞ *2e7*	0/15
U-D	∞	∞	∞	∞	∞	∞ *2e7*	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
P-0	2.3e4(3e4)	∞	∞	∞	∞	∞ *2e7*	0/15
P-D	**2687**(7228)	∞	∞	∞	∞	∞ *2e7*	0/15
U-0	7416(1e4)	∞	∞	∞	∞	∞ *2e7*	0/15
U-D	2759(7235)	∞	∞	∞	∞	∞ *2e7*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f17	63	1030	4005	30677	56288	80472	15/15
P-0	2.3e4(7)	∞	∞	∞	∞	∞ *2e7*	0/15
P-D	2.6(2)	∞	∞	∞	∞	∞ *2e7*	0/15
U-0	**1.7**(0.7)	∞	∞	∞	∞	∞ *2e7*	0/15
U-D	1.9(1)	∞	∞	∞	∞	∞ *2e7*	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
P-0	∞	∞	∞	∞	∞	∞ *2e7*	0/15
P-D	∞	∞	∞	∞	∞	∞ *2e7*	0/15
U-0	7.5e4(8e4)	∞	∞	∞	∞	∞ *2e7*	0/15
U-D	**1.7e4**(2e4)	∞	∞	∞	∞	∞ *2e7*	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
P-0	3.2e7(4e7)	∞	∞	∞	∞	∞ *2e7*	0/15
P-D	4.3e6(1e7)	∞	∞	∞	∞	∞ *2e7*	0/15
U-0	**317**(200)	∞	∞	∞	∞	∞ *2e7*	0/15
U-D	397(212)	**2.9e8**(3e8)	∞	∞	∞	∞ *2e7*	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
P-0	22(5)	**77**(145)	∞	∞	∞	∞ *2e7*	0/15
P-D	22(5)	11(14)	∞	∞	∞	∞ *2e7*	0/15
U-0	**9.4**(4)	**0.32**(0.3)	∞	∞	∞	∞ *2e7*	0/15
U-D	16(6)	1.4(0.6)	**93**(103)	**52**(56)	**51**(56)	**51**(53)	1/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f21	561	6541	14103	14643	15567	17589	15/15
P-0	**5484**(2e4)	2.0e4(2e4)	**9218**(1e4)	**8878**(1e4)	**8351**(9636)	**7392**(9096)	2/15
P-D	2.4e4(4e4)	4.3e4(5e4)	2.0e4(2e4)	1.9e4(2e4)	1.8e4(2e4)	1.6e4(2e4)	1/15
U-0	1.8e4(4e4)	**2.0e4**(2e4)	9218(1e4)	8880(1e4)	8356(1e4)	7407(9096)	2/15
U-D	8910(2e4)	2.0e4(2e4)	9219(1e4)	8882(1e4)	8362(1e4)	7414(9094)	2/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f22	467	5580	23491	24948	26847	1.3e5	12/15
P-0	2.9e4(4e4)	2.3e4(3e4)	∞	∞	∞	∞ *2e7*	0/15
P-D	3.7e4(4e4)	**2.3e4**(3e4)	∞	∞	∞	∞ *2e7*	0/15
U-0	**2.1e4**(4e4)	∞	∞	∞	∞	∞ *2e7*	0/15
U-D	2.9e4(4e4)	∞	∞	∞	∞	∞ *2e7*	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
P-0	2.1(2)	2304(5644)	∞	∞	∞	∞ *2e7*	0/15
P-D	**2.1**(2)	**1083**(876)	∞	∞	∞	∞ *2e7*	0/15
U-0	2.4(3)	3936(6286)	∞	∞	∞	∞ *2e7*	0/15
U-D	2.1(2)	4711(6313)	∞	∞	∞	∞ *2e7*	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
P-0	∞	∞	∞	∞	∞	∞ *2e7*	0/15
P-D	∞	∞	∞	∞	∞	∞ *2e7*	0/15
U-0	∞	∞	∞	∞	∞	∞ *2e7*	0/15
U-D	∞	∞	∞	∞	∞	∞ *2e7*	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 $f_{\mathrm{opt}} + 10^{-8}$. Best results are printed in bold. Algorithm names are abbreviated (e.g. P-0 is P-zero).