Black-Box Optimization Benchmarking: Comparison of Two PSO_Bounds Algorithms on the Noiseless Testbed

Draft version *

Mohammed El-Abd
American University of Kuwait
melabd@auk.edu.kw

ABSTRACT

This paper benchmarks the updated PSO_Bounds algorithm using the noise-free BBOB 2010 testbed in comparison with the original PSO_Bounds algorithm benchmarked in 2009.

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

1. ALGORITHM PRESENTATION

A population-based incremental learning (PBIL) approach for continuous search spaces was proposed in [8]. The algorithm explored the search space by dividing the domain of each gene into two equal intervals referred to as the *low* and *high* intervals. A probability h_d, which is initially set to 0.5, is the probability of dimension number d being in the *high* interval as shown:

$$x_d \in [a, b], h_d = Probability(x_d > \frac{a+b}{2}) \quad (1)$$

After each generation, this distribution was updated according to the dimension values of the best individual using the following formula:

$$p = \begin{cases} 0 & \texttt{if} \quad x_d^{best} < \frac{a+b}{2} \\ 1 & otherwise \end{cases} \quad (2)$$

$$h_d^{t+1} = (1-\alpha) * h_d^t + \alpha * p$$

where α is the *relaxation factor* and t is the iteration number. If h_d gets below h_{dmin} or above h_{dmax}, the population gets re-sampled in the corresponding interval, $[a, \frac{a+b}{2}]$ or $[\frac{a+b}{2}, b]$ respectively.

El-Abd and Kamel [1] introduced PSO_Bounds, in which the concepts of population-based incremental learning (PBIL)are integrated into PSO. At the beginning of the algorithm, the particles are initialized in the predefined domain. After every iteration, the probability h_d of each dimension d is adjusted according to the probability of the value associated with this dimension being in the *high* interval of the defined domain. To prevent premature convergence, this probability is calculated using information from all the particles and not only *gbest*. Hence, the original equations of PBIL are changed as follows:

$$p_{id}^t = \begin{cases} 0 & \texttt{if} \quad pbest_{id}^t < \frac{a+b}{2} \\ 1 & otherwise \end{cases}$$

$$p_d^t = \frac{\sum_i^n p_{id}^t}{n}$$

$$h_d^{t+1} = (1-\alpha) * h_d^t + \alpha * p_d^t \quad (3)$$

where $i \in \{1..n\}$ and n is the number of particles, t is the iteration number, and d is the dimension.

In PBIL, the probabilities were updated using the value of the best individual, which is analogous to the current position of the particles in PSO. However, in our implementation, we use the values of *pbest* instead because these values reflect the best experience of the swarm and would guide the search towards better solutions. When h_d becomes specific enough, the domain of dimension d is adjusted accordingly, and h_d is re-initialized to 0.5. In this model, different dimensions may end up having different domains and different velocity bounds which do not happen in normal PSO.

In order to overcome the problem of the bounds overlapping, thus preventing further particle movement, the width of the adjusted bounds is taken into consideration if the algorithm needs to adjust these bounds for a certain dimension d. If the width drops below a predetermined percentage of the initial search domain width, controlled by the parameter T, the bounds are reset to the initial bounds of the search space and the velocity component is also re-initialized. This will allow the particles to move in different directions and in large steps in the next iteration while still taking the old *pbest* and *gbest* information into account, hence, not losing any previous information gathered during the search.

Previous experimentation in [1, 2] showed that the performance of the algorithm is highly dependent on the val-

*Submission deadline: March 25th.

GECCO'10, July 7–11, 2010, Portland Oregon, United States of America.

ues of the tuple <h_{dmin}, h_{dmax}, α>. Values <0.1,0.9,0.01> (providing a slow update) are better for uni-modal functions while <0.2,0.8,0.1> (providing a fast update) are better for multi-modal functions. In [2], the values were set as <0.2,0.8,0.05> to have a moderate behavior.

2. UPDATED ALGORITHM

In this algorithm, a set of different parameter tunings are defined as P1=<0.1,0.9,0.01>, P2=<0.2,0.8,0.05> and P3=<0.2,0.8,0.1>. Initially, all dimensions have a slow update (following the setting P1).

If the width of a certain dimension drops below the predetermined threshold, the bounds are reset to the initial bounds of the search space, the velocity component is also re-initialized, and in addition, the update speed is changed to P2. If the width of the same dimension drops below the threshold again, the same thing happens while changing the parameter settings to P3. In short, every time the width of a certain dimension drops below the predetermined threshold, re-initialization techniques adopted in the original PSO_Bounds algorithm are applied while changing the update speed as well.

The updated PSO_Bounds algorithm is allowed to perform a maximum of $10^5 \times D$ function evaluations for all test functions and no restart mechanism was used.

3. RESULTS

Results from experiments according to [4] on the benchmark functions given in [3, 5] are presented in Figures 1, 2 and 3 and in Table 1. The **expected running time (ERT)** depends on a given target function value, $f_\mathrm{t} = f_\mathrm{opt} + \Delta f$, 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 [4, 7].

Statistical significance is tested with the rank-sum test for a given target Δf_t (10^{-8} 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 achieved Δf value.

4. CPU TIMING EXPERIMENT

For the timing experiment, the updated PSO_Bounds was run on f8 and restarted until at least 30 seconds had passed (according to Figure 2 in [6]). The experiments have been conducted with an Intel Core 2 Quad 2.4 GHz under Windows Vista using the MATLAB-code provided. The results were $1.2, 1.2, 1.5, 1.7$ and 2.0×10^{-5} seconds per function evaluation in dimensions 2, 3, 5, 10 and 20, respectively. An increase in CPU time with the search space dimensionality is detected.

5. REFERENCES

[1] M. El-Abd and M. S. Kamel. Particle swarm optimization with varying bounds. In *IEEE Congress on Evolutionary Computation*, pages 4757–4761, 2007.

[2] M. El-Abd and M. S. Kamel. Black-box optimization benchmarking for noiseless function testbed using pso_bounds. In *GECCO (Companion)*, pages 2275–2280, 2009.

[3] 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.

[4] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Technical Report RR-7215, INRIA, 2010.

[5] 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.

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

[7] 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.

[8] I. Servet, L. Trave-Massuyes, and D. Stern. Telephone network traffic overloading diagnosis and evolutionary computation technique. In *Artificial Evolution. Springer-Verlag, LNCS 1363*, pages 137–144, 1997.

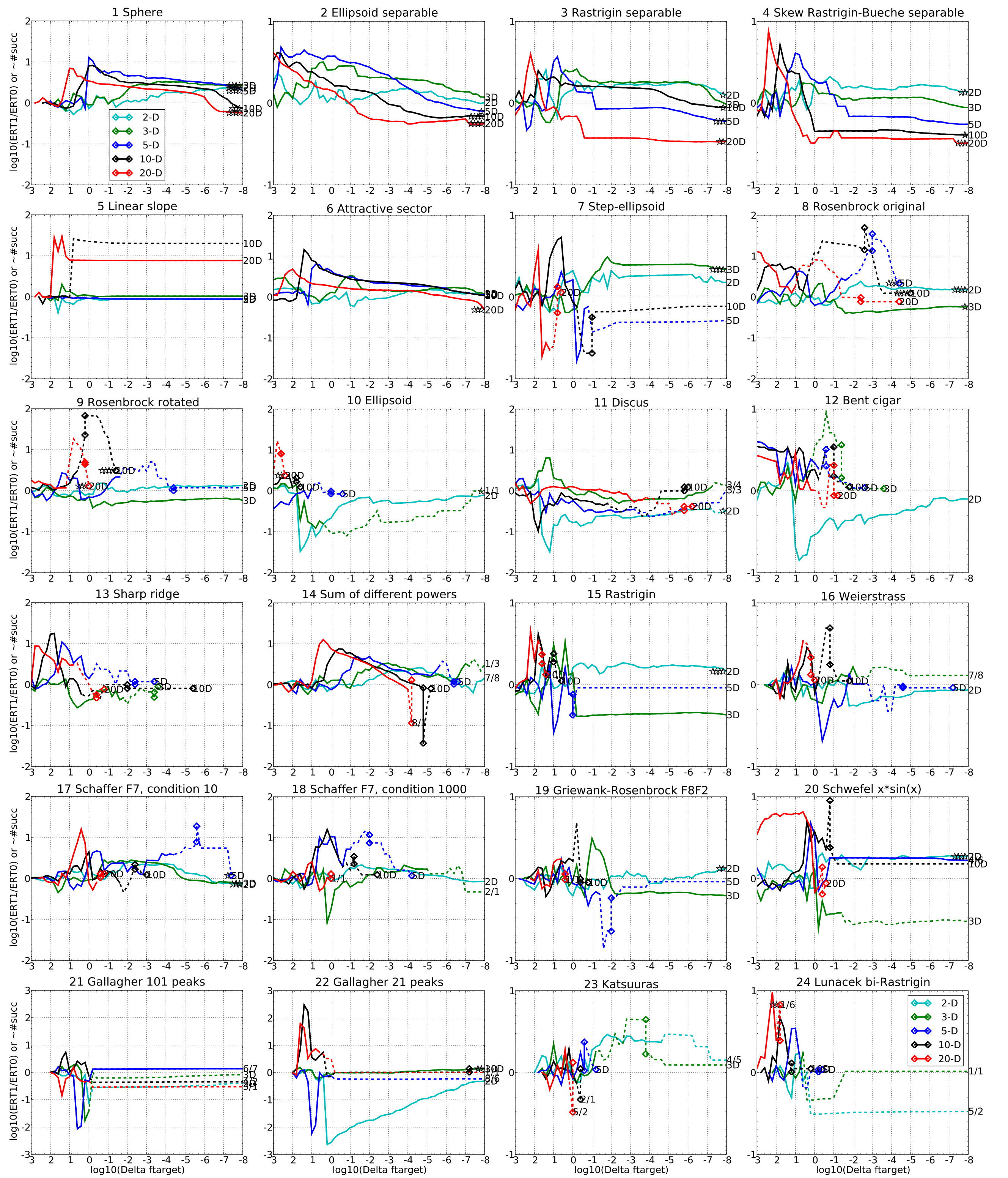

Figure 1: ERT ratio of ALG1-acronym divided by ALG0-acronym versus $\log_{10}(\Delta f)$ for f_1–f_{24} in 2, 3, 5, 10, 20, 40-D. Ratios $< 10^0$ indicate an advantage of ALG1-acronym, smaller values are always better. The line gets dashed when for any algorithm the ERT exceeds thrice the median of the trial-wise overall number of f-evaluations for the same algorithm on this function. Symbols indicate the best achieved Δf-value of one algorithm (ERT gets undefined to the right). The dashed line continues as the fraction of successful trials of the other algorithm, where 0 means 0% and the y-axis limits mean 100%, values below zero for ALG1-acronym. The line ends when no algorithm reaches Δf anymore. The number of successful trials is given, only if it was in $\{1 \dots 9\}$ for ALG1-acronym (1st number) and non-zero for ALG0-acronym (2nd number). Results are significant with $p = 0.05$ for one star and $p = 10^{-\#\star}$ otherwise, with Bonferroni correction within each figure.

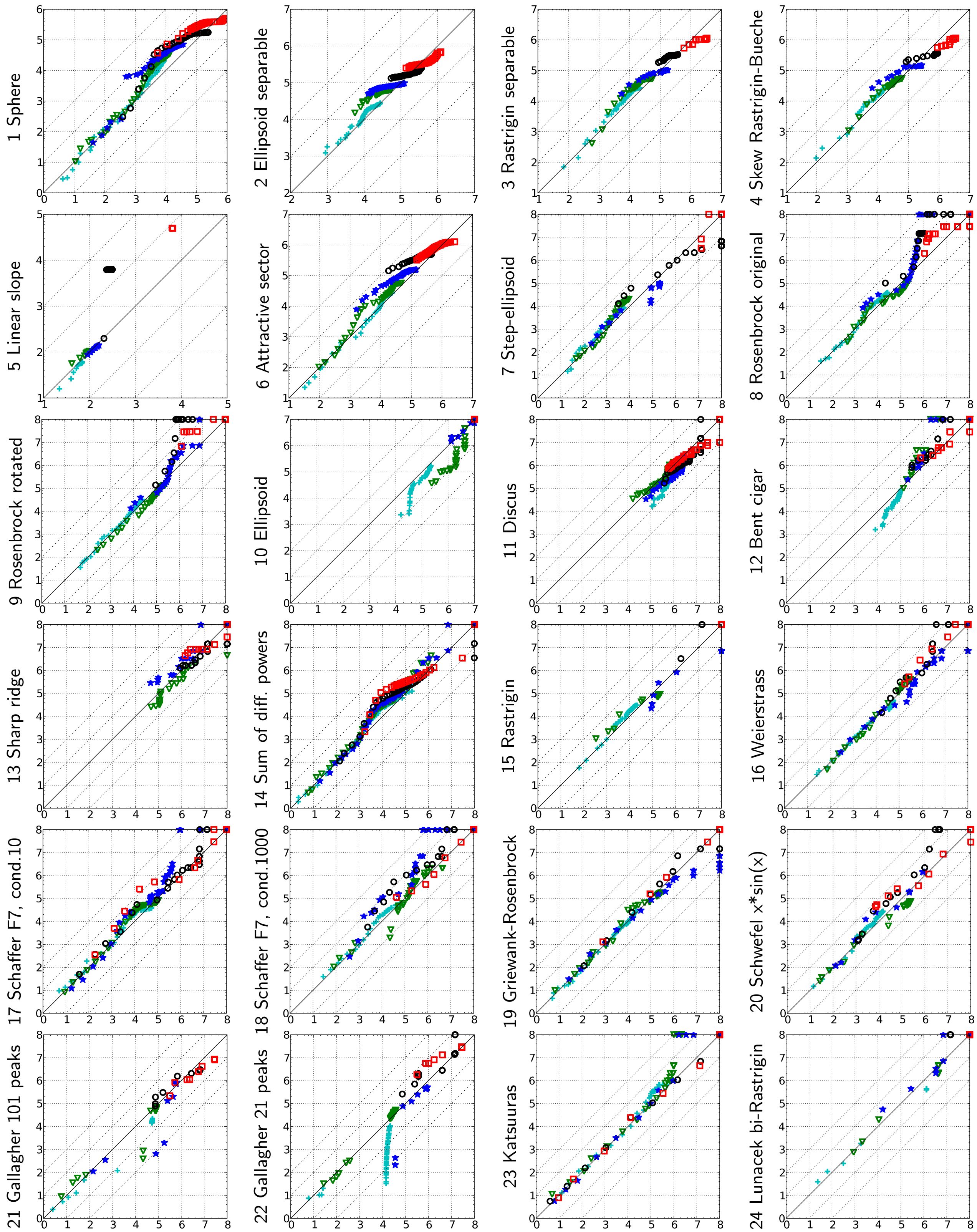

Figure 2: Expected running time (ERT in log10 of number of function evaluations) of ALG1-acronym versus ALG0-acronym for 46 target values $\Delta f \in [10^{-8}, 10]$ in each dimension for functions f_1–f_{24}. Markers on the upper or right egde indicate that the target value was never reached by ALG1-acronym or ALG0-acronym respectively. Markers represent dimension: 2:+, 3:▽, 5:⋆, 10:○, 20:□, 40:◇.

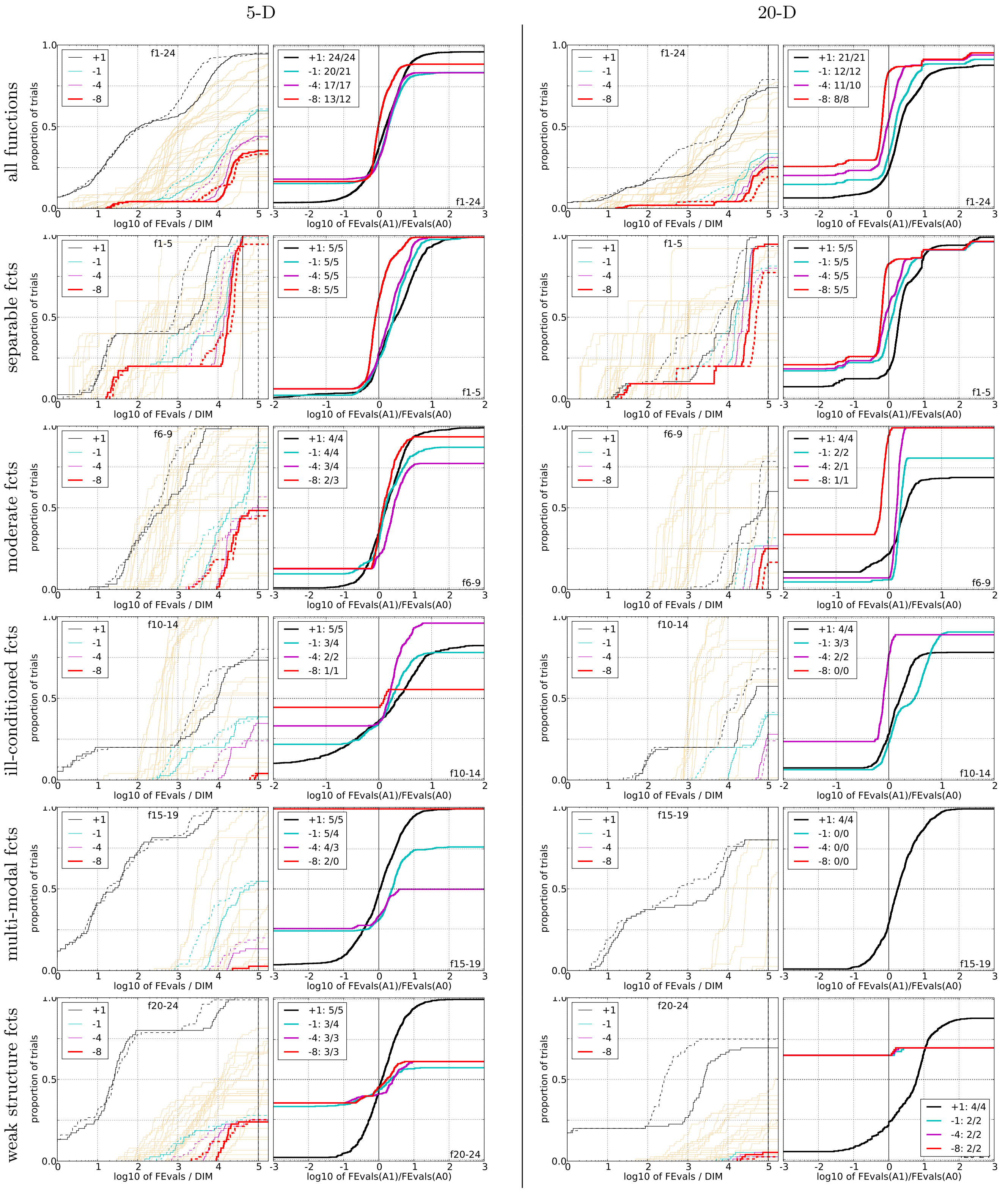

Figure 3: Empirical cumulative distributions (ECDF) plotting a fraction of trials in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of necessary function evaluations divided by dimension D (FEvals/D) to reached a target value $f_{\mathrm{opt}}+\Delta f$ with $\Delta f = 10^k$, where $k \in \{1, -1, -4, -8\}$ is given by the first value in the legend, for ALG1-acronym (solid) and ALG0-acronym (dashed). Light beige lines show the ECDF of FEvals for target value $\Delta f = 10^{-8}$ of all algorithms benchmarked during BBOB-2009. Right sub-columns: ECDF of FEval ratios between all trial pairs of ALG1-acronym divided by ALG0-acronym. Pairs where both trials failed are disregarded, pairs where one trial failed are visible in the limits being > 0 or < 1. The legends indicate the number of functions that were solved in at least one trial (ALG1-acronym first).

5-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f1	11	12	12	12	12	12	15/15
0: PB	3.8	41	**210***	**730**$^{\star 3}$	**1.3e3**$^{\star 3}$	**1.9e3**$^{\star 3}$	15/15
1: UPB	4.1	520	1.1e3	2.9e3	4.4e3	5.4e3	15/15
f2	83	87	88	90	92	94	15/15
0: PB	**150**$^{\star 2}$	**190**$^{\star 3}$	**260**$^{\star 3}$	**400**$^{\star 3}$	860	1.2e3	15/15
1: UPB	600	720	800	870	950	1.0e3	15/15
f3	720	1600	1600	1600	1700	1700	15/15
0: PB	**7.6**$^{\star 2}$	**26**$^{\star 2}$	38	64	70	95	14/15
1: UPB	24	47	51	55	59	62	15/15
f4	810	1600	1700	1800	1900	1900	15/15
0: PB	**8**$^{\star 3}$	**30**$^{\star 2}$	64	110	110	140	12/15
1: UPB	33	57	81	78	76	77	15/15
f5	10	10	10	10	10	10	15/15
0: PB	9.2	15	16	16	16	16	15/15
1: UPB	8.8	13	14	14	14	14	15/15
f6	110	210	280	580	1000	1300	15/15
0: PB	14	49	85	98	78	92	15/15
1: UPB	69	220	260	210	140	120	15/15
f7	24	320	1200	1600	1600	1600	15/15
0: PB	9.4	13	170	130	130	130	11/15
1: UPB	10	20	61	65	65	65	14/15
f8	73	270	340	390	410	420	15/15
0: PB	**30***	470	920	**1.4e3***	∞	∞ *5.0e5*	0/15
1: UPB	120	290	1.3e3	1.9e4	∞	∞ *5.0e5*	0/15
f9	35	130	210	300	340	370	15/15
0: PB	220	1.5e3	1.6e3	4.0e3	2.2e4	2.0e4	1/15
1: UPB	390	1.0e3	2.2e3	1.2e4	∞	∞ *5.0e5*	0/15
f10	350	500	570	630	830	880	15/15
0: PB	3.8e3	1.5e4	∞	∞	∞	∞ *5.0e5*	0/15
1: UPB	4.2e3	1.5e4	∞	∞	∞	∞ *5.0e5*	0/15
f11	140	200	760	1200	1500	1700	15/15
0: PB	430	1.4e3	1.0e3	1.1e3	1.5e3	1.4e3	3/15
1: UPB	240	830	340	370	470	700	3/15
f12	110	270	370	460	1300	1500	15/15
0: PB	1.9e3	2.5e3	5.5e3	∞	∞	∞ *5.0e5*	0/15
1: UPB	2.3e3	5.4e3	∞	∞	∞	∞ *5.0e5*	0/15
f13	130	190	250	1300	1800	2300	15/15
0: PB	**350**$^{\star 2}$	2.4e3	5.7e3	5.4e3	∞	∞ *5.0e5*	0/15
1: UPB	2.2e3	3.3e3	1.3e4	∞	∞	∞ *5.0e5*	0/15
f14	9.8	41	58	140	250	480	15/15
0: PB	1.9	12	**45***	**140**$^{\star 3}$	410	∞ *5.0e5*	0/15
1: UPB	1.6	9.3	100	360	710	∞ *5.0e5*	0/15
f15	510	9300	1.9e4	2.0e4	2.1e4	2.1e4	14/15
0: PB	170	120	∞	∞	∞	∞ *5.0e5*	0/15
1: UPB	44	91	370	350	340	330	1/15
f16	120	610	2700	1.0e4	1.2e4	1.2e4	15/15
0: PB	2.4	36	140	320	∞	∞ *5.0e5*	0/15
1: UPB	2.3	37	94	200	610	590	0/15
f17	5.2	210	900	3700	6400	7900	15/15
0: PB	3.4	7.9	**52**$^{\star 2}$	51	61	120	0/15
1: UPB	2.4	18	110	140	350	∞ *5.0e5*	0/15
f18	100	380	4000	9300	1.1e4	1.2e4	15/15
0: PB	3.8	**21**$^{\star 3}$	69	120	∞	∞ *5.0e5*	0/15
1: UPB	2.9	230	360	∞	∞	∞ *5.0e5*	0/15
f19	1	1	240	1.2e5	1.2e5	1.2e5	15/15
0: PB	27	1.6e4	2.5e3	∞	∞	∞ *5.0e5*	0/15
1: UPB	31	1.4e4	1.6e3	30	60	60	1/15
f20	16	850	3.8e4	5.4e4	5.5e4	5.5e4	14/15
0: PB	8.1	**8.6**$^{\star 3}$	21	15	15	16	6/15
1: UPB	7.4	30	38	27	27	27	4/15
f21	41	1200	1700	1700	1700	1800	14/15
0: PB	3.5	380	340	340	340	340	7/15
1: UPB	2.8	180	450	450	450	450	6/15
f22	71	390	940	1000	1000	1100	14/15
0: PB	510	870	820	810	820	820	6/15
1: UPB	3	650	470	470	470	480	8/15
f23	3	520	1.4e4	3.2e4	3.3e4	3.4e4	15/15
0: PB	2.1	58	240	∞	∞	∞ *5.0e5*	0/15
1: UPB	1.9	49	∞	∞	∞	∞ *5.0e5*	0/15
f24	1600	2.2e5	6.4e6	9.6e6	1.3e7	1.3e7	3/15
0: PB	**10**$^{\star 3}$	33	∞	∞	∞	∞ *5.0e5*	0/15
1: UPB	35	33	∞	∞	∞	∞ *5.0e5*	0/15

20-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f1	43	43	43	43	43	43	15/15
0: PB	**120**$^{\star 3}$	**1.5e3**$^{\star 3}$	**2.1e3**$^{\star 3}$	**3.3e3**$^{\star 3}$	**4.5e3**$^{\star 3}$	1.6e4	15/15
1: UPB	860	5.1e3	6.1e3	7.3e3	8.3e3	**9.9e3**$^{\star 3}$	15/15
f2	380	390	390	390	390	390	15/15
0: PB	**360**$^{\star 2}$	**530***	840	1.8e3	2.3e3	2.6e3	14/15
1: UPB	650	750	810	**1.1e3**$^{\star 3}$	**1.3e3**$^{\star 3}$	**1.6e3**$^{\star 3}$	15/15
f3	5100	7600	7600	7600	7600	7700	15/15
0: PB	120	190	360	360	400	430	7/15
1: UPB	110	130	130	140	140	**140***	13/15
f4	4700	7600	7700	7700	7800	1.4e5	9/15
0: PB	190	290	300	360	380	21	7/15
1: UPB	120	92	130	140	**140**$^{\star 2}$	**7.9**$^{\star 2}$	13/15
f5	41	41	41	41	41	41	15/15
0: PB	160	160	160	160	160	160	15/15
1: UPB	1.2e3	1.2e3	1.2e3	1.2e3	1.2e3	1.2e3	15/15
f6	1300	2300	3400	5200	6700	8400	15/15
0: PB	**120**$^{\star 3}$	**110**$^{\star 3}$	**100**$^{\star 3}$	**96**$^{\star 3}$	**110**$^{\star 2}$	160	10/15
1: UPB	250	200	170	140	140	140	15/15
f7	1400	4300	9500	1.7e4	1.7e4	1.7e4	15/15
0: PB	9.7e3	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	2.4e3	∞	∞	∞	∞	∞ *2.0e6*	0/15
f8	2000	3900	4000	4200	4400	4500	15/15
0: PB	520	**430***	**1.8e3***	∞	∞	∞ *2.0e6*	0/15
1: UPB	970	3.5e3	7.2e3	6.9e3	∞	∞ *2.0e6*	0/15
f9	1700	3100	3300	3500	3600	3700	15/15
0: PB	**700***	**9.7e3**$^{\star 2}$	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	3.9e3	∞	∞	∞	∞	∞ *2.0e6*	0/15
f10	7400	8700	1.1e4	1.5e4	1.7e4	1.7e4	15/15
0: PB	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
f11	1000	2200	6300	9800	1.2e4	1.5e4	15/15
0: PB	570	440	220	480	1.2e3	∞ *2.0e6*	0/15
1: UPB	730	550	240	300	590	∞ *2.0e6*	0/15
f12	1000	1900	2700	4100	1.2e4	1.4e4	15/15
0: PB	700	3.0e3	5.1e3	∞	∞	∞ *2.0e6*	0/15
1: UPB	2.0e3	3.0e3	1.0e4	∞	∞	∞ *2.0e6*	0/15
f13	650	2000	2800	1.9e4	2.4e4	3.0e4	15/15
0: PB	2.3e3	4.1e3	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	6.6e3	4.1e3	∞	∞	∞	∞ *2.0e6*	0/15
f14	75	240	300	930	1600	1.6e4	15/15
0: PB	23	**100**$^{\star 3}$	**170**$^{\star 3}$	**380**$^{\star 3}$	∞	∞ *2.0e6*	0/15
1: UPB	28	770	830	550	∞	∞ *2.0e6*	0/15
f15	3.0e4	1.5e5	3.1e5	3.2e5	4.5e5	4.6e5	15/15
0: PB	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
f16	1400	2.7e4	7.7e4	1.9e5	2.0e5	2.2e5	15/15
0: PB	130	1.0e3	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	190	∞	∞	∞	∞	∞ *2.0e6*	0/15
f17	63	1000	4000	3.1e4	5.6e4	8.0e4	15/15
0: PB	3	830	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	5.5	650	∞	∞	∞	∞ *2.0e6*	0/15
f18	620	4000	2.0e4	6.8e4	1.3e5	1.5e5	15/15
0: PB	**69***	7.1e3	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	180	7.2e3	∞	∞	∞	∞ *2.0e6*	0/15
f19	1	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
0: PB	820	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	1.3e3	∞	∞	∞	∞	∞ *2.0e6*	0/15
f20	82	4.6e4	3.1e6	5.5e6	5.6e6	5.6e6	14/15
0: PB	**86**$^{\star 3}$	11	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	540	7.7	∞	∞	∞	∞ *2.0e6*	0/15
f21	560	6500	1.4e4	1.5e4	1.6e4	1.8e4	15/15
0: PB	560	1.2e3	2.0e3	1.9e3	1.8e3	1.6e3	1/15
1: UPB	390	640	580	570	540	480	3/15
f22	470	5600	2.3e4	2.5e4	2.7e4	1.3e5	12/15
0: PB	**680***	730	1.2e3	1.1e3	1.1e3	210	1/15
1: UPB	3.9e3	2.4e3	1.2e3	1.1e3	1.1e3	220	1/15
f23	3.2	1600	6.7e4	4.9e5	8.1e5	8.4e5	15/15
0: PB	3	8.4e3	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	2.5	2.8e3	∞	∞	∞	∞ *2.0e6*	0/15
f24	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
0: PB	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: UPB	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15

Table 1: Expected running time (ERT in number of function evaluations) divided by the best ERT measured during BBOB-2009 (given in the respective first row) for different Δf values for functions f_1–f_{24}. #succ is the number of trials that reached the final target $f_{\rm opt} + 10^{-8}$. 0: PB is ALG0-acronym and 1: UPB is ALG1-acronym. Results are statistically significant compared to the best in 2009 with $p = 0.05$ for one star and $p = 10^{-x}$ otherwise, where x is the figure next to the star, with Bonferroni correction of 48.