Probability Matching-based Adaptive Strategy Selection vs. Uniform Strategy Selection within Differential Evolution

An Empirical Comparison on the BBOB-2010 Noiseless Testbed

Álvaro Fialho
Microsoft Research – INRIA Joint Centre
Parc Orsay Université
91893 Orsay, France
alvaro.fialho@inria.fr

Wenyin Gong
School of Computer Science
China University of Geosciences
Wuhan, 430074 P.R. China
cug11100304@yahoo.com.cn

Zhihua Cai
School of Computer Science
China University of Geosciences
Wuhan, 430074 P.R. China
zhcai@cug.edu.cn

ABSTRACT

Different strategies can be used for the generation of new candidate solutions on the Differential Evolution algorithm. However, the definition of which of them should be applied to the problem at hand is not trivial, besides being a sensitive choice with relation to the algorithm performance. In this paper, we use the BBOB-2010 noiseless benchmarking suite to further empirically validate the Probability Matching-based Adaptive Strategy Selection (*PM-AdapSS-DE*) [4], a method proposed to automatically select the mutation strategy to be applied, based on the relative fitness improvements recently achieved by the application of each of the available strategies on the current optimization process. It is compared with what would be a timeless (naïve) choice, the uniform strategy selection within the same sub-set of strategies.

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, Adaptive Strategy Selection, Differential Evolution.

1. INTRODUCTION

Differential Evolution (DE) is a simple yet powerful evolutionary algorithm, that uses the weighted difference between two or more candidate solutions to generate a new one. Proposed by Storn and Price [13], the DE algorithm was initially provided with a single mutation strategy for its offspring generation. Ten other strategies were lately suggested by the same authors [11], increasing the robustness of the algorithm with relation to many different application domains, such as data mining, pattern recognition, digital filter design, neural network training, etc.

But, although bringing advantages, such flexibility creates an extra difficulty to the user. Given an optimization problem, there are no definitive guidelines or rules-of-thumb to assist the user into the definition of which strategy should be used. Such critical choice is usually done following the intuition, or by means of statistics over an expensive set of experiments (off-line tuning).

The latter approach can be always used to find the best strategy; however, besides being computationally and timely expensive, it will very probably yield a sub-optimal choice. Intuitively, a subsequent use of strategies with different characteristics during the optimization process should achieve better performance, as the search tends to migrate from a global (early) exploration of the search space to a more focused, exploitation-like behavior.

This is the main motivation for the use of adaptive parameter control methods, that automatically selects the strategy that should be applied for the generation of each offspring while solving the problem, based on their recent performance on the current optimization process, what is referred to as *Adaptive Strategy Selection*.

In this paper, the BBOB-2010 noiseless benchmarking suite is used to further empirically validate a technique recently proposed to do so, the *PM-AdapSS-DE* [4]. For the sake of self-containedness, the description of the proposed adaptive method is resumed in Section 2, referring the reader to the original paper [4] for more a complete and detailed view. The rest of the paper is organized as follows. The settings used for the empirical comparison are presented in Section 3. The complete results are shown in Section 4, while Section 5 presents the timing complexity of each algorithm. Some final considerations conclude the paper in Section 6.

2. THE PM-ADAPSS-DE ALGORITHM

Inspired by some recent works in the Genetic Algorithms community (see, *e.g.*, [14, 1]), *Adaptive Strategy Selection* aims at automatically selecting between the available (possibly ill-known) mutation strategies while solving the prob-

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.

lem. Two components are needed to achieve such objective: the *Credit Assignment* scheme defines how to assess the performance of each strategy, measuring the impact of its applications on the progress of the current search/optimization process; while the *Strategy* (or Operator) *Selection* mechanism sets how the strategies are chosen, based on their known performance.

In the *PM-AdapSS-DE* algorithm, the relative fitness improvement η_i, proposed in [9], was adopted to assess the performance of each strategy application, which can be formalized as follows:

$$\eta_i = \frac{\delta}{cf_i} \cdot |pf_i - cf_i| \tag{1}$$

where $i = 1, \cdots, NP$, being NP the population size, δ the fitness of the best-so-far solution in the population, and pf_i and cf_i, the fitness of the target parent and of its offspring, respectively. In case of no improvement (*i.e.*, the offspring is worse than or equal to its target parent), a null reward is awarded. Finally, the credit assigned to each strategy is the absolute average value (referred to as *AvgAbs* in the original paper [4]) of the rewards received by them during a given period (a generation in this case).

This credit is used by the Probability Matching (PM) technique to update the known empirical quality of each strategy. Its impact on the update of this estimate is weighted by a user-defined parameter, the adaptation rate $\alpha \in]0, 1]$ [3]. The probability of selection of each strategy is then defined proportionally to its known performance, possibly being lower bounded by another user-defined parameter, the minimal probability p_{min}, which ensures that no strategy gets lost during the process [14].

The coupling of such elements with the DE algorithm, referred to as the *PM-AdapSS-DE*, is shown in Algorithm 1, reproduced from [4]. The modified steps with respect to the classical DE algorithm are marked with a left arrow "⇐". Summarizing, at each generation t, for each target parent i, a strategy SI_i is selected based on the probability of each strategy. The offspring is generated with such strategy, and its impact (the relative fitness improvement η_i) is calculated and stored in the set S_{SI_i}. Consequently, at the end of each generation, the assigned credit, quality, and probability of each strategy are updated.

3. EXPERIMENTAL SETTINGS

Given that the objective of the present work is to validate the ability of the proposed *Adaptive Strategy Selection* approach, a sub-set of 4 strategies was arbitrarily chosen, listed as follows.

1) "DE/rand/1": $\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \big(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\big)$
2) "DE/rand/2": $\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \big(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\big) + F \cdot \big(\mathbf{x}_{r_4} - \mathbf{x}_{r_5}\big)$
3) "DE/rand-to-best/2": $\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \big(\mathbf{x}_{best} - \mathbf{x}_{r_1}\big) + F \cdot \big(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\big) + F \cdot \big(\mathbf{x}_{r_4} - \mathbf{x}_{r_5}\big)$
4) "DE/current-to-rand/1": $\mathbf{v}_i = \mathbf{x}_i + F \cdot \big(\mathbf{x}_{r_1} - \mathbf{x}_i\big) + F \cdot \big(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\big)$

where $\mathbf{x}_i$ represents the current individual, $\mathbf{x}_{best}$ is the best individual in the current generation, r_1, r_2, r_3, r_4, r_5 are individuals randomly chosen from the population, being $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$. F is the mutation scaling factor.

Algorithm 1 DE with probability matching-based adaptive strategy selection: *PM-AdapSS-DE*

Set $CR = 1.0, F = 0.5$ and $NP = 10 \times D$
Generate the initial population
Evaluate the fitness for each individual
Set the generation counter $t = 1$
Set $K = 4, p_{min} = 0$, and $\alpha = 0.6$ ⇐
For each strategy a, set $q_a(t) = 0$ and $p_a(t) = 1/K$ ⇐
while The halting criterion is not satisfied **do**
  for $i = 1$ to NP **do**
    Select the strategy SI_i based on its probability ⇐
    Select uniform randomly $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$
    $j_{rand} = \text{rndint}(1, D)$
    for $j = 1$ to D **do**
      if $\text{rndreal}_j[0, 1) < CR$ **or** $j == j_{rand}$ **then**
        if $SI_i == 1$ **then**
          $u_{i,j}$ is generated by "DE/rand/1" strategy
        else if $SI_i == 2$ **then**
          $u_{i,j}$ is generated by "DE/rand/2" strategy
        else if $SI_i == 3$ **then**
          $u_{i,j}$ is generated by "DE/rand-to-best/2" strategy
        else if $SI_i == 4$ **then**
          $u_{i,j}$ is generated by "DE/current-to-rand/1"
        end if
      else
        $u_{i,j} = x_{i,j}$
      end if
    end for
  end for
  for $i = 1$ to NP **do**
    Evaluate the offspring $\mathbf{u}_i$
    if $f(\mathbf{u}_i)$ is better than **or** equal to $f(\mathbf{x}_i)$ **then**
      Calculate η_i using Eqn. (1) ⇐
      Replace $\mathbf{x}_i$ with $\mathbf{u}_i$
    else
      Set $\eta_i = 0$ ⇐
    end if
    $S_{SI_i} \leftarrow \eta_i$ ⇐
  end for
  Calculate the credit $r_a(t)$ for each strategy ⇐
  Update the quality $q_a(t)$ for each strategy ⇐
  Update the probability $p_a(t)$ for each strategy ⇐
  $t = t + 1$
end while

The parameters of the PM technique were defined after a preliminary tuning phase. For the adaptation rate, the following values were tried: $\alpha \in \{.1, .3, .6, .9\}$; while for the minimal probability $p_{min} \in \{0, .05, .1, .2\}$. Each of the sixteen configurations composed by the combination of such parameter values was tried once on all the functions and instances for dimensions 5 and 20 (a kind of representative set of all the analyzed dimensions, summing up to 720 instances), and the best configuration ($p_{min} = 0; \alpha = .6$) was found, according to the Friedman's two-way Analysis of Variances by Ranks statistical test.

The user-defined parameters of the DE algorithm, namely the population size NP and the mutation scaling factor F were defined, respectively, to $10 * D$, and 0.5, with D being the dimensionality of the problem. Differently from the original paper [4], $CR = 1.0$ was used here, in order to have a DE invariant with relation to rotation, and entirely dependent on the mutation strategy application [7].

The technique used as baseline for comparison represents what would be a possible (and costless) choice for a naïve user, *i.e.*, the uniform selection within the same sub-set of strategies. In the original paper [4], the *PM-AdapSS-DE* is also compared to a DE implementing each of the strategies alone, and to yet another adaptive scheme, known as SaDE [12] (with fixed CR and F in this case).

The experiments were performed following the BBOB guidelines [5], with the maximum number of evaluations being fixed at $10^5 * D$. The mentioned parameter values were used on all the experiments, for all dimensions, thus crafting effort is equal to zero.

4. RESULTS

Results from experiments according to [5] on the benchmark functions given in [2, 6] are presented in Figures 1, 2 and 3 and in Table 1. The **expected running time (ERT)**, used in the figures and table, 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} [5, 10]. **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 Δf-value achieved, measured only up to the smallest number of overall function evaluations for any unsuccessful trial under consideration.

5. CPU TIMING EXPERIMENTS

For the timing experiments, both algorithms were run on f8 and restarted until at least 30 seconds (according to [5]). The experiments have been conducted with an Intel Xeon E5345 processor (2.33 GHz) running Linux 2.6.31.12. For the *PM-AdapSS-DE*, the results were 1.6, 1.6, 1.6, 1.8, 2.1 and 2.9 $\times 10^{-6}$ seconds per function evaluation, for the dimensions 2, 3, 5, 10, 20, 40 respectively. For the baseline technique, the *Uniform-DE*, the results were 3.3, 3.5 4.0, 5.5, 8.3 and 16 $\times 10^{-7}$ seconds per function evaluation, for the dimensions 2, 3, 5, 10, 20, 40 respectively. The same C++ implementation (*gcc* version 4.4.1) was used for both, with the only difference being the portions of code that refer to the strategy selection, thus the timing difference shows exactly the price to be paid for using this adaptive scheme.

6. FINAL CONSIDERATIONS

This work presented a more extensive empirical validation of the *PM-AdapSS-DE*, recently proposed in [4]. This method provides *Adaptive Strategy Selection* capabilities to the DE algorithm, by means of the *Probability Matching* strategy selection scheme, that selects the operators according to the *relative fitness improvements* brought by their recent applications.

The objective of this work was not to compete with the state-of-the-art continuous optimizers, but rather to analyze the advantages brought by the adaptive method when compared to the naïve (uniform) choice. Although in the multimodal and weak-structure group of functions it was less clear, improvements were achieved over the baseline method in most of the functions, especially for the larger dimensions, being statistically equivalent otherwise. A possible explanation for the fewer successes concerning the smaller dimensions might lie in the fact that those experiments were too short to allow the adaptive scheme to learn and show its skills. The ill-conditioned functions were the ones in which the advantages of the adaptive scheme was better shown.

There is still a lot of space for further improvements, especially concerning the strategy selection mechanism. Other schemes, such as the Adaptive Pursuit [14] and the Bandit-based approaches [1] should be analyzed. Another path that could also be explored lies in the automatic adaptation of other DE parameters, the crossover rate CR and the mutation scaling factor F, as proposed for the SaDE [12] scheme. Besides, especially concerning the multi-modal functions, the diversity could also be considered somehow by the credit assignment scheme, as proposed in [8].

7. REFERENCES

[1] A. Fialho, M. Schoenauer, and M. Sebag. Analysis of adaptive operator selection techniques on the royal road and long k-path problems. In *Proc. GECCO*, pages 779–786, 2009.

[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] D. E. Goldberg. Probability matching, the magnitude of reinforcement, and classifier system bidding. *Mach. Learn.*, 5(4):407–425, 1990.

[4] W. Gong, A. Fialho, and Z. Cai. Adaptive strategy selection in differential evolution. In J. Branke et al., editor, *Proc. GECCO*. ACM Press, 2010.

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

[7] A. W. Iorio and X. Li. Improving the performance and scalability of differential evolution. In *Proc. SEAL*, pages 131–140. Springer-Verlag, 2008.

[8] J. Maturana, F. Lardeux, and F. Saubion. Autonomous operator management for evolutionary algorithms. *J. of Heuristics*, 2010.

[9] Y.-S. Ong and A. J. Keane. Meta-Lamarckian learning in memetic algorithms. *IEEE Trans. on Evol. Comput.*, 8(2):99–110, Apr 2004.

[10] K. Price. Differential evolution vs. the functions of the second ICEO. In *Proc. CEC*, pages 153–157, 1997.

[11] K. Price, R. Storn, and J. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer-Verlag, Berlin, 2005.

[12] A. K. Qin, V. L. Huang, and P. N. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. on Evol. Comput.*, 13(2):398–417, Apr 2009.

[13] R. Storn and K. Price. Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optim.*, 11(4):341–359, Dec 1997.

[14] D. Thierens. An adaptive pursuit strategy for allocating operator probabilities. In *Proc. GECCO*, pages 1539–1546, 2005.

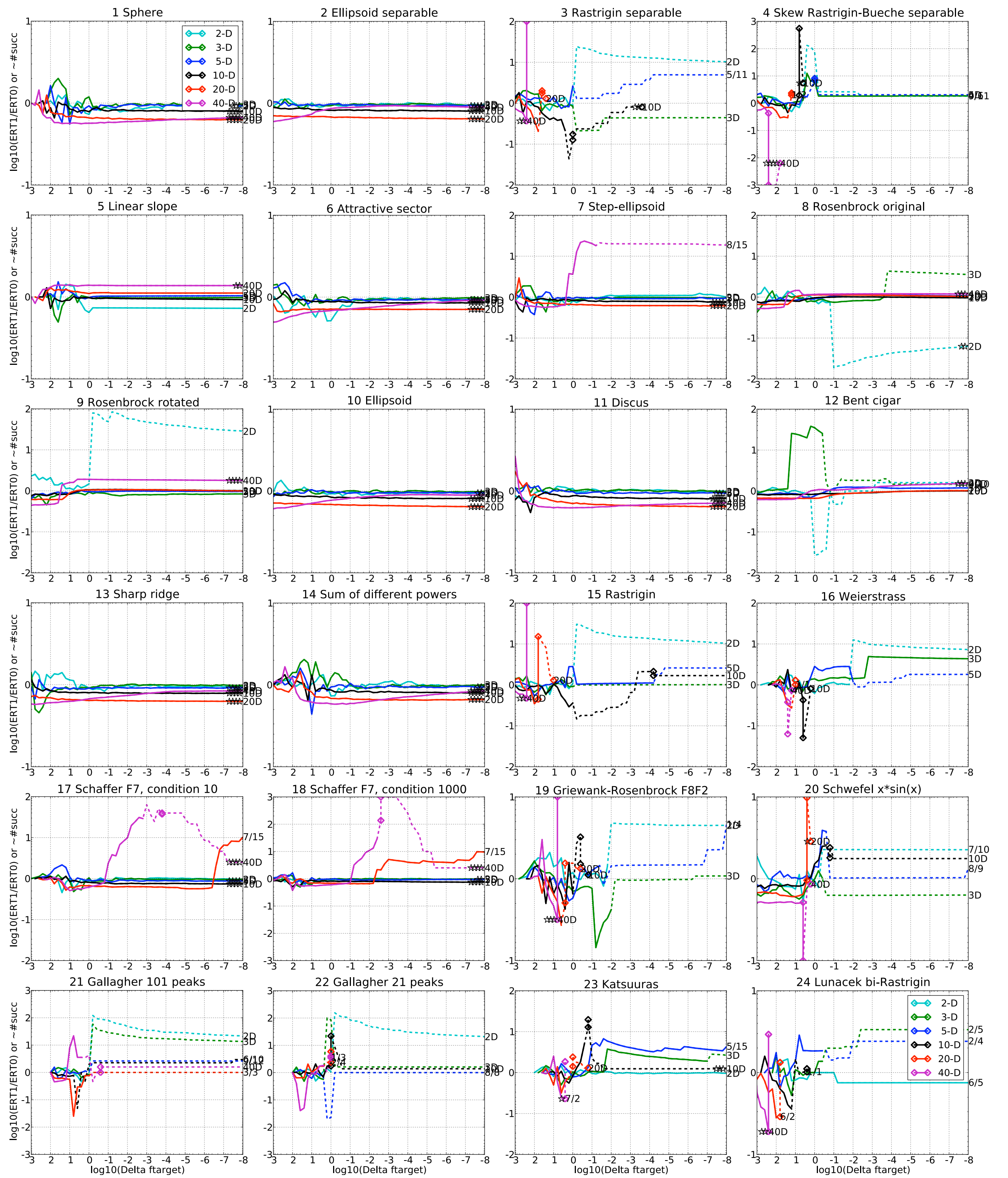

Figure 1: ERT ratio of PM-AdapSS-DE divided by Uniform-DE versus $\log_{10}(\Delta f)$ for f_1–f_{24} in 2, 3, 5, 10, 20, 40-D. Ratios $< 10^0$ indicate an advantage of PM-AdapSS-DE, smaller values are always better. The line gets dashed when for any algorithm the ERT exceeds thrice the median of the 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 PM-AdapSS-DE. The line ends when no algorithm reaches Δf anymore. The number of successful trials is given, only if it was in $\{1 \ldots 9\}$ for PM-AdapSS-DE (1st number) and non-zero for Uniform-DE (2nd number). Results are significant with $p = 0.05$ for one star and $p = 10^{-\#\star}$ otherwise, with Bonferroni correction within each figure.

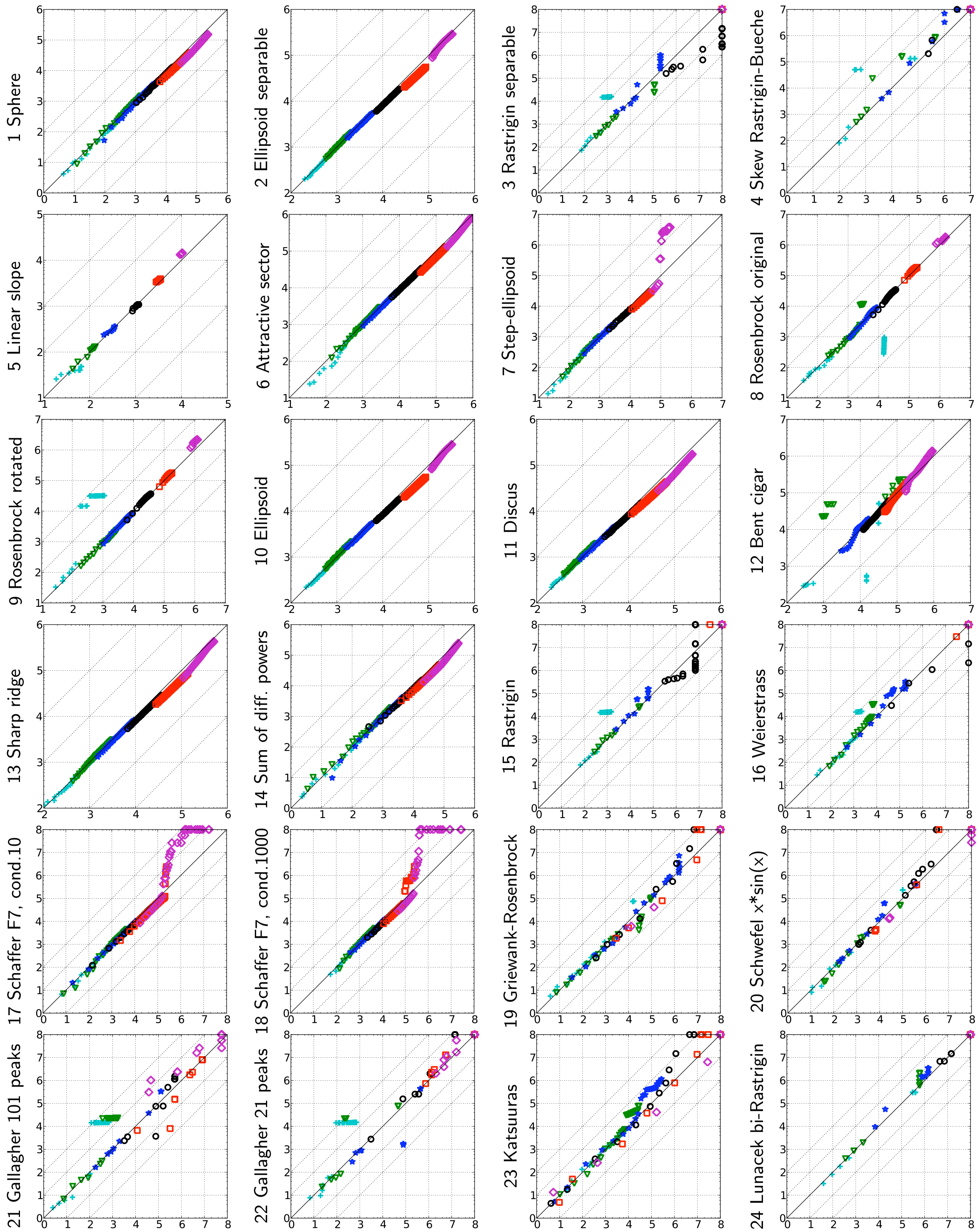

Figure 2: Expected running time (ERT in log10 of number of function evaluations) of PM-AdapSS-DE versus Uniform-DE for 46 target values $\Delta f \in [10^{-8}, 10]$ in each dimension for functions f_1–f_{24}. Markers on the upper or right egde indicate that the target value was never reached by PM-AdapSS-DE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

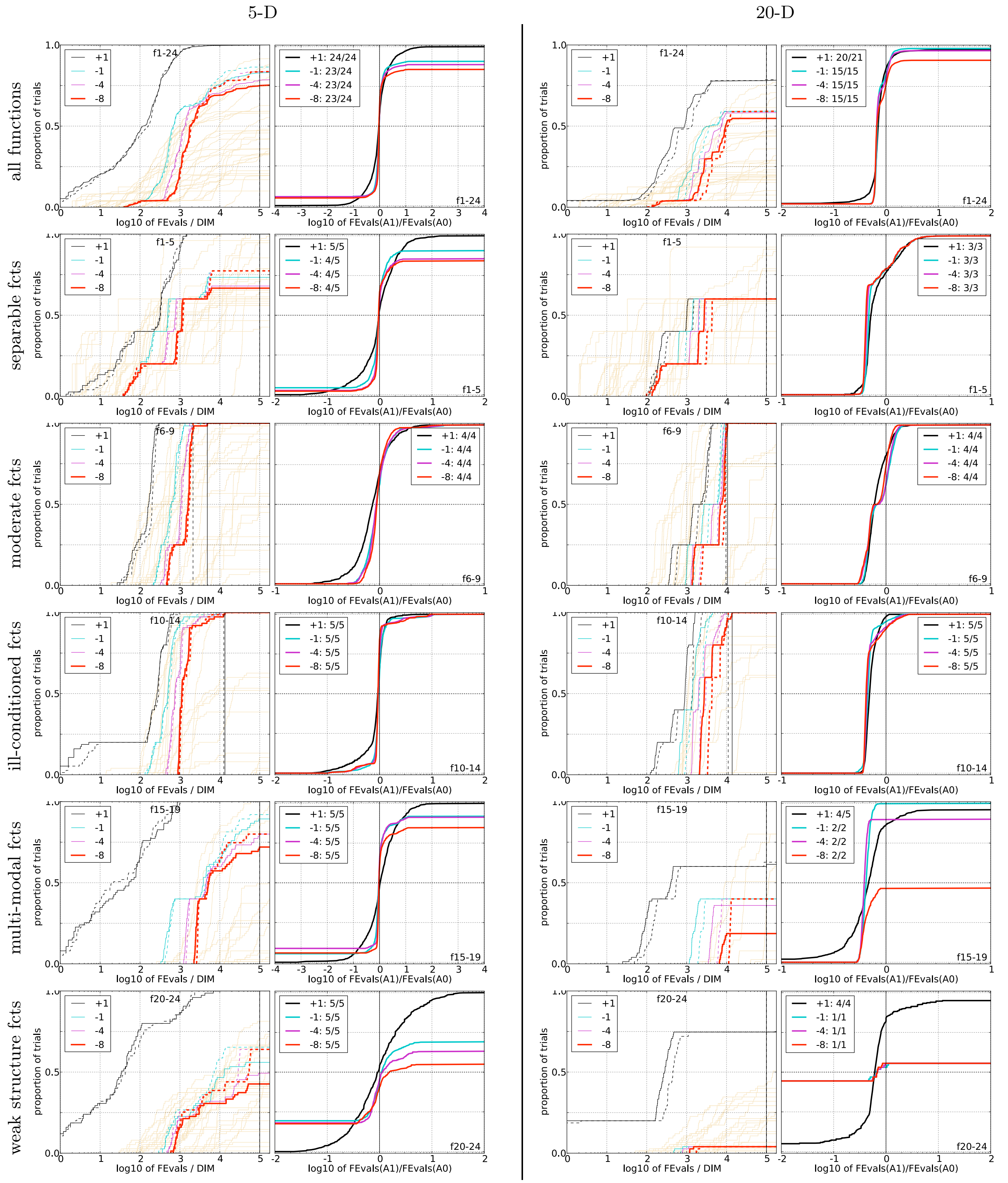

Figure 3: Empirical cumulative distributions (ECDF) of run lengths and speed-up ratios in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of function evaluations divided by dimension D (FEvals/D) to reach a target value $f_{\mathrm{opt}} + \Delta f$ with $\Delta f = 10^k$, where $k \in \{1, -1, -4, -8\}$ is given by the first value in the legend, for PM-AdapSS-DE (solid) and Uniform-DE (dashed). Light beige lines show the ECDF of FEvals for target value $\Delta f = 10^{-8}$ of algorithms benchmarked during BBOB-2009. Right sub-columns: ECDF of FEval ratios of PM-AdapSS-DE divided by Uniform-DE, all trial pairs for each function. 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 (PM-AdapSS-DE first).

5-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f_1	11	12	12	12	12	12	15/15
0: Uniform	8.6	42	80	160	240	310	15/15
1: AdapSS	4.8	39	74	140	220	**290**$^{\star 2}$	15/15
f_2	83	87	88	90	92	94	15/15
0: Uniform	20	24	30	40	50	58	15/15
1: AdapSS	19	24	28	38	**47***	**55**$^{\star 2}$	15/15
f_3	720	1600	1600	1600	1700	1700	15/15
0: Uniform	3.5	12	130	130	130	130	11/15
1: AdapSS	4.9	33	170	360	620	620	5/15
f_4	810	1600	1700	1800	1900	1900	15/15
0: Uniform	5.1	630	1.9e3	1.8e3	1.7e3	1.7e3	2/15
1: AdapSS	4.9	4.3e3	∞	∞	∞	∞ *5.0e5*	0/15
f_5	10	10	10	10	10	10	15/15
0: Uniform	20	34	35	35	35	35	15/15
1: AdapSS	24	33	36	36	36	36	15/15
f_6	110	210	280	580	1000	1300	15/15
0: Uniform	9	9.2	10	7.8	6.1	6.1	15/15
1: AdapSS	8.1	8.1	9	7.3	5.7	**5.7**$^{\star 2}$	15/15
f_7	24	320	1200	1600	1600	1600	15/15
0: Uniform	14	2.7	1.2	1.4	1.4	1.6	15/15
1: AdapSS	12	2.5	1.2	1.3	1.3	1.5	15/15
f_8	73	270	340	390	410	420	15/15
0: Uniform	15	10	14	17	18	20	15/15
1: AdapSS	13	9.4	15	19	20	21	15/15
f_9	35	130	210	300	340	370	15/15
0: Uniform	30	23	21	21	22	22	15/15
1: AdapSS	25	22	21	21	21	22	15/15
f_{10}	350	500	570	630	830	880	15/15
0: Uniform	4.5	4.2	4.6	5.5	5.4	6.2	15/15
1: AdapSS	4.6	4	4.3	5.3	5.2	5.9	15/15
f_{11}	140	200	760	1200	1500	1700	15/15
0: Uniform	6.2	6.6	2.4	2.4	2.6	2.8	15/15
1: AdapSS	6.3	6.5	2.3	2.3	**2.4***	**2.7**$^{\star 2}$	15/15
f_{12}	110	270	370	460	1300	1500	15/15
0: Uniform	28	21	21	22	10	11	15/15
1: AdapSS	24	16	24	27	12	13	15/15
f_{13}	130	190	250	1300	1800	2300	15/15
0: Uniform	11	12	13	3.7	3.8	3.7	15/15
1: AdapSS	**9.9***	**11***	**12***	**3.4**$^{\star 3}$	**3.5**$^{\star 3}$	**3.4**$^{\star 3}$	15/15
f_{14}	9.8	41	58	140	250	480	15/15
0: Uniform	2.3	10	16	15	13	9.3	15/15
1: AdapSS	1	9.4	17	15	12	8.9	15/15
f_{15}	510	9300	1.9e4	2.0e4	2.1e4	2.1e4	14/15
0: Uniform	4.8	2.2	3	3	2.9	2.9	14/15
1: AdapSS	5.4	6.3	3.2	3.2	7.6	7.4	12/15
f_{16}	120	610	2700	1.0e4	1.2e4	1.2e4	15/15
0: Uniform	4.3	44	20	17	16	15	12/15
1: AdapSS	3.9	120	55	20	28	27	11/15
f_{17}	5.2	210	900	3700	6400	7900	15/15
0: Uniform	3.8	4.3	2.7	1.6	1.5	1.6	15/15
1: AdapSS	4.2	4	**2.3***	**1.4**$^{\star 2}$	**1.3**$^{\star 3}$	**1.4**$^{\star 3}$	15/15
f_{18}	100	380	4000	9300	1.1e4	1.2e4	15/15
0: Uniform	4	4.3	0.81	0.73	0.95	1.1	15/15
1: AdapSS	4.3	4.1	0.76	0.67	0.89	1	15/15
f_{19}	1	1	240	1.2e5	1.2e5	1.2e5	15/15
0: Uniform	35	3.4e3	1.6e3	13	13	13	4/15
1: AdapSS	37	2.0e3	2.1e3	19	19	29	1/15
f_{20}	16	850	3.8e4	5.4e4	5.5e4	5.5e4	14/15
0: Uniform	11	10	9.2	6.4	6.4	6.4	9/15
1: AdapSS	11	14	9.4	6.6	6.6	6.6	8/15
f_{21}	41	1200	1700	1700	1700	1800	14/15
0: Uniform	4.5	33	76	75	74	74	12/15
1: AdapSS	4	33	200	200	200	190	9/15
f_{22}	71	390	940	1000	1000	1100	14/15
0: Uniform	6.6	200	470	440	420	410	8/15
1: AdapSS	4.1	4.6	470	440	420	410	8/15
f_{23}	3	520	1.4e4	3.2e4	3.3e4	3.4e4	15/15
0: Uniform	2	11	2.5	3.6	5.5	7.2	15/15
1: AdapSS	1.8	8.9	12	14	21	27	5/15
f_{24}	1600	2.2e5	6.4e6	9.6e6	1.3e7	1.3e7	3/15
0: Uniform	4.3	3.7	0.17	0.15	0.11	0.11	4/15
1: AdapSS	5.9	6.8	0.24	0.36	0.27	0.27	2/15

20-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f_1	43	43	43	43	43	43	15/15
0: Uniform	150	300	440	730	1.0e3	1.3e3	15/15
1: AdapSS	**100**$^{\star 3}$	**200**$^{\star 3}$	**290**$^{\star 3}$	**470**$^{\star 3}$	**650**$^{\star 3}$	**830**$^{\star 3}$	15/15
f_2	380	390	390	390	390	390	15/15
0: Uniform	77	92	110	140	170	200	15/15
1: AdapSS	**52**$^{\star 3}$	**63**$^{\star 3}$	**73**$^{\star 3}$	**93**$^{\star 3}$	**110**$^{\star 3}$	**130**$^{\star 3}$	15/15
f_3	5100	7600	7600	7600	7600	7700	15/15
0: Uniform	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: AdapSS	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_4	4700	7600	7700	7700	7800	1.4e5	9/15
0: Uniform	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: AdapSS	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_5	41	41	41	41	41	41	15/15
0: Uniform	69	83	85	86	86	86	15/15
1: AdapSS	82	92	96	96	96	96	15/15
f_6	1300	2300	3400	5200	6700	8400	15/15
0: Uniform	29	23	20	20	20	20	15/15
1: AdapSS	**20**$^{\star 3}$	**16**$^{\star 3}$	**15**$^{\star 3}$	**14**$^{\star 3}$	**14**$^{\star 3}$	**14**$^{\star 3}$	15/15
f_7	1400	4300	9500	1.7e4	1.7e4	1.7e4	15/15
0: Uniform	8.4	5.2	3.3	2.7	2.7	2.7	15/15
1: AdapSS	**5.9**$^{\star 3}$	**3.4**$^{\star 3}$	**2.1**$^{\star 3}$	**1.6**$^{\star 3}$	**1.6**$^{\star 3}$	**1.7**$^{\star 3}$	15/15
f_8	2000	3900	4000	4200	4400	4500	15/15
0: Uniform	34	31	33	35	37	39	15/15
1: AdapSS	35	35	38	39	40	41	15/15
f_9	1700	3100	3300	3500	3600	3700	15/15
0: Uniform	41	39	41	43	45	46	15/15
1: AdapSS	37	41	43	45	46	47	15/15
f_{10}	7400	8700	1.1e4	1.5e4	1.7e4	1.7e4	15/15
0: Uniform	3.9	4.1	3.9	3.6	3.9	4.6	15/15
1: AdapSS	**2.7**$^{\star 3}$	**2.8**$^{\star 3}$	**2.6**$^{\star 3}$	**2.4**$^{\star 3}$	**2.6**$^{\star 3}$	**3**$^{\star 3}$	15/15
f_{11}	1000	2200	6300	9800	1.2e4	1.5e4	15/15
0: Uniform	11	8.1	3.8	3.8	4	4.2	15/15
1: AdapSS	**8.7**$^{\star 3}$	**5.7**$^{\star 3}$	**2.6**$^{\star 3}$	**2.5**$^{\star 3}$	**2.6**$^{\star 3}$	**2.7**$^{\star 3}$	15/15
f_{12}	1000	1900	2700	4100	1.2e4	1.4e4	15/15
0: Uniform	44	27	25	26	12	13	15/15
1: AdapSS	**29**$^{\star 3}$	**18**$^{\star 3}$	20	24	12	13	15/15
f_{13}	650	2000	2800	1.9e4	2.4e4	3.0e4	15/15
0: Uniform	43	20	19	4.1	4.2	4.2	15/15
1: AdapSS	**28**$^{\star 3}$	**13**$^{\star 3}$	**12**$^{\star 3}$	**2.6**$^{\star 3}$	**2.6**$^{\star 3}$	**2.6**$^{\star 3}$	15/15
f_{14}	75	240	300	930	1600	1.6e4	15/15
0: Uniform	53	50	64	38	31	4.2	15/15
1: AdapSS	**43**$^{\star 2}$	**34**$^{\star 3}$	**43**$^{\star 3}$	**25**$^{\star 3}$	**20**$^{\star 3}$	**2.8**$^{\star 3}$	15/15
f_{15}	3.0e4	1.5e5	3.1e5	3.2e5	4.5e5	4.6e5	15/15
0: Uniform	980	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: AdapSS	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_{16}	1400	2.7e4	7.7e4	1.9e5	2.0e5	2.2e5	15/15
0: Uniform	2.1e4	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: AdapSS	2.2e4	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_{17}	63	1000	4000	3.1e4	5.6e4	8.0e4	15/15
0: Uniform	38	19	10	3	3.2	2.8	15/15
1: AdapSS	23	**13**$^{\star 3}$	**6.6**$^{\star 3}$	**1.9**$^{\star 3}$	**1.8**$^{\star 3}$	18	7/15
f_{18}	620	4000	2.0e4	6.8e4	1.3e5	1.5e5	15/15
0: Uniform	18	7.8	2.8	1.7	1.5	1.6	15/15
1: AdapSS	**13**$^{\star 3}$	**5.4**$^{\star 3}$	**1.9**$^{\star 3}$	8.4	6.5	7.8	7/15
f_{19}	1	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
0: Uniform	2.8e3	9.5e6	∞	∞	∞	∞ *2.0e6*	0/15
1: AdapSS	**1.8e3***	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_{20}	82	4.6e4	3.1e6	5.5e6	5.6e6	5.6e6	14/15
0: Uniform	76	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: AdapSS	**46**$^{\star 3}$	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_{21}	560	6500	1.4e4	1.5e4	1.6e4	1.8e4	15/15
0: Uniform	21	460	570	550	520	460	3/15
1: AdapSS	**12**$^{\star 3}$	350	570	550	520	460	3/15
f_{22}	470	5600	2.3e4	2.5e4	2.7e4	1.3e5	12/15
0: Uniform	1.6e3	990	∞	∞	∞	∞ *2.0e6*	0/15
1: AdapSS	1.6e3	2.3e3	∞	∞	∞	∞ *2.0e6*	0/15
f_{23}	3.2	1600	6.7e4	4.9e5	8.1e5	8.4e5	15/15
0: Uniform	3	6.0e3	∞	∞	∞	∞ *2.0e6*	0/15
1: AdapSS	1.5	8.5e3	∞	∞	∞	∞ *2.0e6*	0/15
f_{24}	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
0: Uniform	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
1: AdapSS	∞	∞	∞	∞	∞	∞ *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}. 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 $f_{\mathrm{opt}} + 10^{-8}$. 0: Uniform is Uniform-DE and 1: AdapSS is PM-AdapSS-DE. Bold entries are statistically significantly better compared to the other algorithm, with $p = 0.05$ or $p = 10^{-k}$ where $k > 1$ is the number following the $\star$ symbol, with Bonferroni correction of 48.