Comparison of Cauchy EDA and BIPOP-CMA-ES Algorithms on the BBOB Noiseless Testbed

Petr Pošík
Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics
Technická 2, 166 27 Prague 6, Czech Republic
posik@labe.felk.cvut.cz

ABSTRACT

Estimation-of-distribution algorithm using Cauchy sampling distribution is compared with the bi-population CMA evolutionary strategy which was one of the best contenders in the black-box optimization benchmarking workshop in 2009. The results clearly indicate that the CMA evolutionary strategy is in all respects a better optimization algorithm than the Cauchy estimation-of-distribution algorithm. This paper compares both algorithms in more detail and adds to the understanding of their key features and differences.

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, Estimation-of-distribution algorithm, Cauchy distribution, Evolutionary strategy, Covariance matrix adaptation

1. INTRODUCTION

The 2010 issue of the black-box optimization benchmarking methodology (BBOB) [3] allows for a detailed comparison of 2 algorithms on the BBOB functions testbed. In this article, two algorithms benchmarked during the BBOB 2009 workshop are further compared. Data for both algorithms were taken from 2009 benchmarking, but the comparison is made using the new BBOB 2010 post-processing scripts and templates. Both algorithms fall into the class of evolutionary optimization algorithms and both algorithms use unimodal distribution as a mean for generating new offspring; however, there are several important differences between them and this paper clearly shows which algorithm is better. The two algorithms selected for the comparison are:

- The bi-population variant of the evolutionary strategy with covariance matrix adaptation (BIPOP-CMA-ES) [2] which belongs to the best algorithms of the BBOB 2009 comparison in terms of speed and success ratio, and thus was selected as the reference algorithm.
- The estimation-of-distribution algorithm (EDA) with Cauchy sampling distribution (Cauchy EDA) [5]. In BBOB 2010, further comparisons of the Cauchy EDA with other algorithms (G3PCX [6] and Rosenbrock's algorithm [7]) are planned, and this article anchors the relative performances of the respective algorithm pairs to one of the best algorithms in BBOB 2009.

In the next section, both algorithms are shortly described and their differences are emphasized. Sec. 3 contains all the results used to compare the algorithms and their discussions. After the presentation of the time demands of both algorithms in Sec. 4, Sec. 5 concludes the paper.

2. ALGORITHM PRESENTATION

The exact descriptions of the algorithms along with the parameter settings can be found in [5] and [2], respectively. Apart of the unimodality of the sampling distribution, the algorithms differ foremost in the following aspects:

- The probabilistic model used in BIPOP-CMA-ES is Gaussian, while the EDA uses Cauchy distribution.
- In BIPOP-CMA-ES, each generation, the Gaussian distribution is *updated* (the recent model parameters explicitly take part in the process of creating new values of model parameters), while in Cauchy EDA all the distribution parameters are *computed from scratch* (and thus a larger population is needed).
- The restart strategy of BIPOP-CMA-ES allows the algorithm to use different population sizes for each restart. In Cauchy EDA, the population size depends only on the dimensionality of the problem being solved. Very often, the BIPOP-CMA-ES population size is much smaller when compared to the population size of Cauchy EDA, which allows the BIPOP-CMA-ES algorithm to converge faster and to be restarted more often.

For both algorithms, the crafting effort CrE= 0.

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

Table 2: The average time demands per function evaluation (in microseconds) of the two compared algorithms.

Dim	2	3	5	10	20	40
BIPOP-CMA-ES	280	240	200	180	180	200
CauchyEDA	51	17	9	9	11	NA

3. RESULTS

Results from experiments according to [3] on the benchmark functions given in [1, 4] 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, $f_t = f_{opt} + \Delta f$, 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} 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.

The most important observation that can be made from Figures 1, 2, and 3 and from Table 1 is that *BIPOP-CMA-ES is simply more reliable (has higher success rates) and typically 1-2 orders of magnitude faster than Cauchy EDA*! BIPOP-CMA-ES outperforms Cauchy EDA for all functions, (virtually) all dimensions and (virtually) all target levels. The few exceptions happen for low dimensional functions (2D or 3D) and for a very narrow range of target levels.

In Fig. 1, we can see rather regular behaviour of the ERT ratios for unimodal[1] functions (1, 2, 5–14). The ERT ratio is often almost constant (between 1 and 100) for a broad range of the target levels.

For multimodal functions (3, 4, 15–24), the Cauchy EDA algorithm does not work well. The long running times allow for a limited number of restarts only, and it is able to solve only low-dimensional versions of some of the multimodal benchmark functions.

4. CPU TIMING EXPERIMENTS

The time requirements of both algorithms are taken from the respective articles, [2] and [5]. The multistart algorithm was run with the maximal number of evaluations set to 10^5, the basic algorithm was restarted for at least 30 seconds. These experiments have been conducted with an Intel dual core T5600 processor with 1.8 GHz under Linux 2.6.27-11 using MATLAB R2008a for BIPOP-CMA-ES, and on Intel Core 2 CPU, T5600, 1.83 GHz, 1 GB RAM with Windows XP SP3 in MATLAB R2007b for Cauchy EDA. The comparison of the average time demands per function evaluation are shown in Table 2.

The differences in the average time needed for function evaluation are caused by the different population sizes. While BIPOP-CMA-ES often uses populations of a few (or a few tens of) individuals, Cauchy EDA needs larger populations which means that the evaluation routine is called less often and can take advantage of the MATLAB matrix processing capabilities to a larger extent.

[1]The Rosenbrock's functions are actually multimodal, but the local optimum does not pose many difficulties to optimization algorithms.

5. CONCLUSIONS

The results indicate that BIPOP-CMA-ES clearly dominates the Cauchy EDA algorithm regardless of the particular optimization conditions. The adaptation scheme used in CMA-ES needs lower population sizes, is thus faster, and allows for more algorithm restarts. For the functions in the testbed, it seems to be better to have fast local optimizer with the possibility to restart it often.

Acknowledgements

The author is supported by the Grant Agency of the Czech Republic with the grant no. 102/08/P094 entitled "Machine learning methods for solution construction in evolutionary algorithms".

6. REFERENCES

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

[2] N. Hansen. Benchmarking a bi-population CMA-ES on the BBOB-2009 function testbed. In *GECCO (Companion)*, pages 2389–2396, 2009.

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

[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] P. Pošík. BBOB-benchmarking a simple estimation-of-distribution algorithm with Cauchy distribution. In *GECCO '09: Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pages 2309–2314, New York, NY, USA, 2009. ACM.

[6] P. Pošík. BBOB-benchmarking the generalized generation gap model with parent centric crossover. In *GECCO '09: Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pages 2321–2328, New York, NY, USA, 2009. ACM.

[7] P. Pošík. BBOB-benchmarking the Rosenbrock's local search algorithm. In *GECCO '09: Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pages 2337–2342, New York, NY, USA, 2009. ACM.

[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.

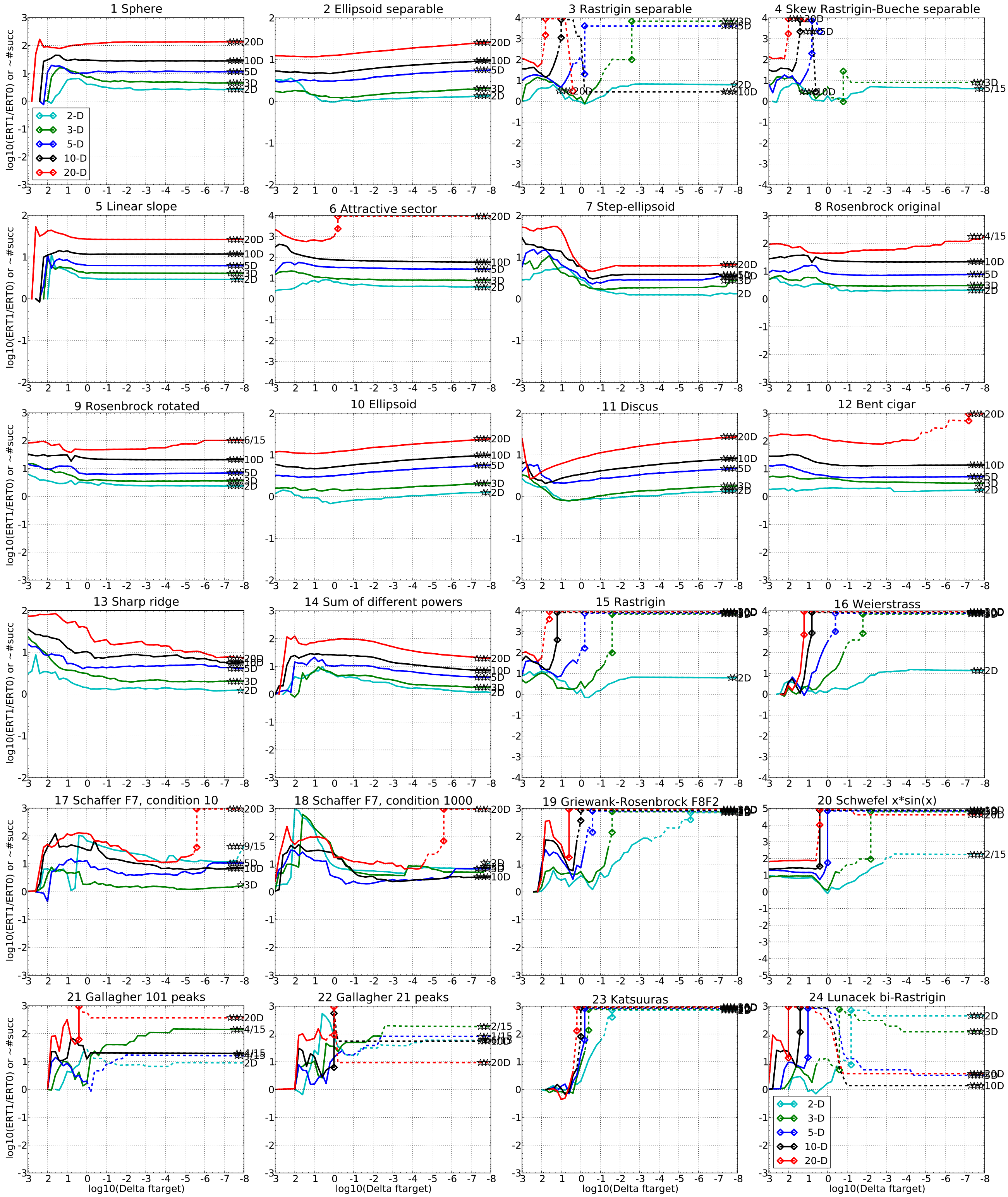

Figure 1: ERT ratio of CauchyEDA divided by BIPOP-CMA-ES versus $\log_{10}(\Delta f)$ for f_1–f_{24} in 2, 3, 5, 10, 20, 40-D. Ratios $< 10^0$ indicate an advantage of CauchyEDA, smaller values are always better. The line gets dashed when for any algorithm the ERT exceeds thrice the median of the trial-wise 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 CauchyEDA. 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 CauchyEDA (1st number) and non-zero for BIPOP-CMA-ES (2nd number). Results are significant with $p = 0.05$ for one star and $p = 10^{-\#\star}$ otherwise, with Bonferroni correction within each figure.

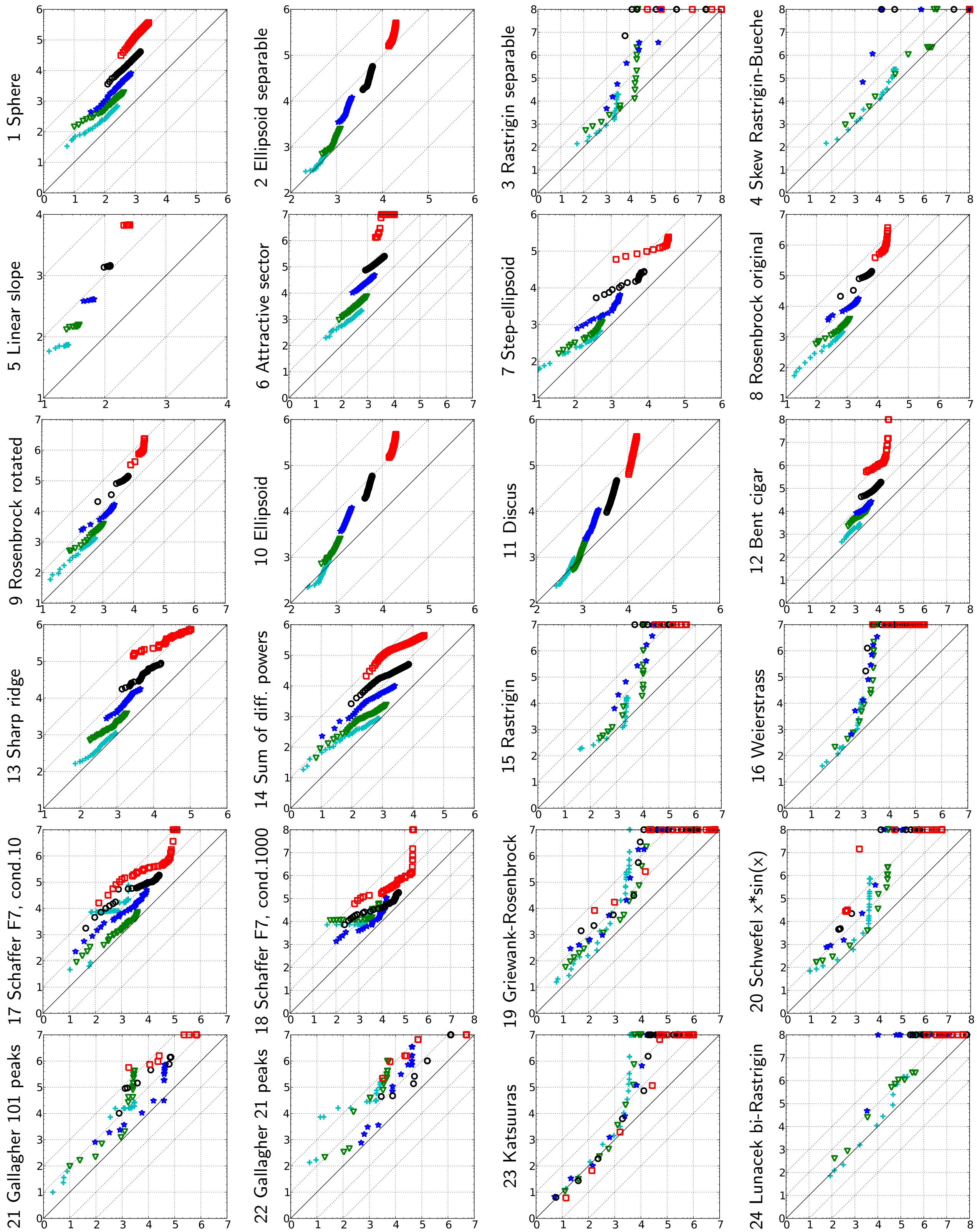

Figure 2: Expected running time (ERT in log10 of number of function evaluations) of CauchyEDA versus BIPOP-CMA-ES for 46 target values $\Delta f \in [10^{-8}, 10]$ in each dimension for functions f_1–f_{24}. Markers on the upper or right egde indicate that the target value was never reached by CauchyEDA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

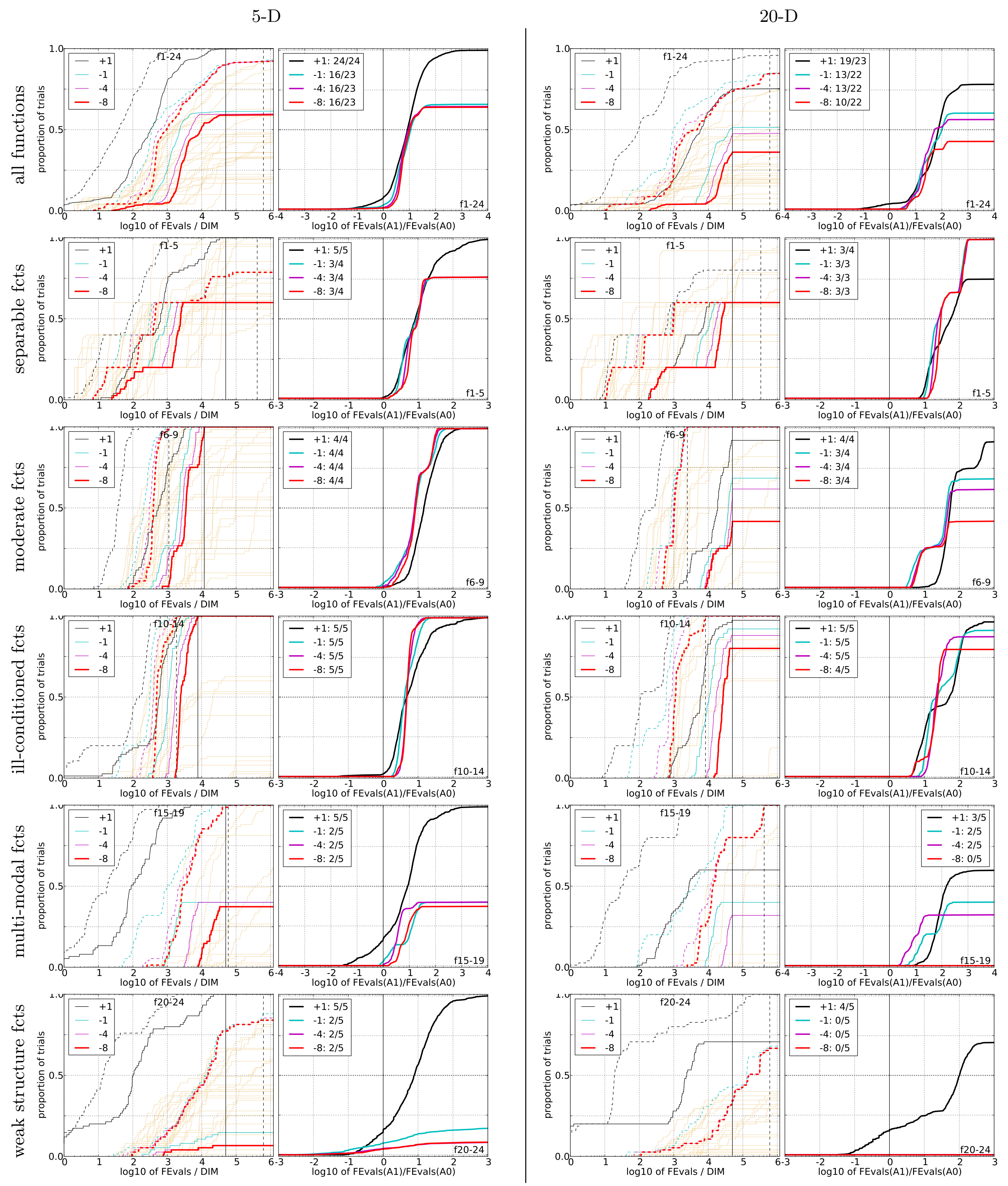

Figure 3: Empirical cumulative distributions (ECDF) of run lengths and speed-up ratios in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of function evaluations divided by dimension D (FEvals/D) to reach a target value $f_{\rm opt} + \Delta f$ with $\Delta f = 10^k$, where $k \in \{1, -1, -4, -8\}$ is given by the first value in the legend, for CauchyEDA (solid) and BIPOP-CMA-ES (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 CauchyEDA divided by BIPOP-CMA-ES, 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 (CauchyEDA first).

5-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
$\mathbf{f_1}$	11	12	12	12	12	12	15/15
0: BIP	**3.2**^{*3}	**9**^{*3}	**15**^{*3}	**27**^{*3}	**40**^{*3}	**53**^{*3}	15/15
1: Cau	41	90	170	310	460	600	15/15
$\mathbf{f_2}$	83	87	88	90	92	94	15/15
0: BIP	**13**^{*3}	**16**^{*3}	**18**^{*3}	**20**^{*3}	**21**^{*3}	**22**^{*3}	15/15
1: Cau	42	49	58	80	100	120	15/15
$\mathbf{f_3}$	720	1600	1600	1600	1700	1700	15/15
0: BIP	**1.4**^{*3}	**16**^{*3}	**140**^{*3}	**140**^{*3}	**140**^{*3}	**140**^{*3}	14/15
1: Cau	6.7	2.2e3	∞	∞	∞	∞ *2.5e5*	0/15
$\mathbf{f_4}$	810	1600	1700	1800	1900	1900	15/15
0: BIP	**2.7**^{*3}	∞	∞	∞	∞	∞ *1.8e6*	0/15
1: Cau	85	∞	∞	∞	∞	∞ *2.5e5*	0/15
$\mathbf{f_5}$	10	10	10	10	10	10	15/15
0: BIP	**4.5**^{*3}	**6.5**^{*3}	**6.6**^{*3}	**6.6**^{*3}	**6.6**^{*3}	**6.6**^{*3}	15/15
1: Cau	39	41	41	41	41	41	15/15
$\mathbf{f_6}$	110	210	280	580	1000	1300	15/15
0: BIP	**2.3**^{*3}	**2.1**^{*3}	**2.2**^{*3}	**1.7**^{*3}	**1.3**^{*3}	**1.3**^{*3}	15/15
1: Cau	92	69	68	47	35	34	15/15
$\mathbf{f_7}$	24	320	1200	1600	1600	1600	15/15
0: BIP	**5**^{*3}	**1.5**^{*3}	**1**^{*2}	**1**^{*3}	**1**^{*3}	**1**^{*3}	15/15
1: Cau	33	4.9	2.4	2.9	2.9	3.4	15/15
$\mathbf{f_8}$	73	270	340	390	410	420	15/15
0: BIP	**3.2**^{*3}	**3.7**^{*3}	**4.5**^{*3}	**4.8**^{*3}	**5.1**^{*3}	**5.4**^{*3}	15/15
1: Cau	49	31	33	34	37	40	15/15
$\mathbf{f_9}$	35	130	210	300	340	370	15/15
0: BIP	**5.8**^{*3}	**8.7**^{*3}	**7.2**^{*3}	**6.4**^{*3}	**6.3**^{*3}	**6.2**^{*3}	15/15
1: Cau	71	54	45	41	42	43	15/15
$\mathbf{f_{10}}$	350	500	570	630	830	880	15/15
0: BIP	**3.5**^{*3}	**2.9**^{*3}	**2.7**^{*3}	**2.8**^{*3}	**2.3**^{*3}	**2.4**^{*3}	15/15
1: Cau	11	9	9.4	12	11	13	15/15
$\mathbf{f_{11}}$	140	200	760	1200	1500	1700	15/15
0: BIP	**8.4**^{*3}	**7.2**^{*3}	**2.2**^{*3}	**1.6**^{*3}	**1.4**^{*3}	**1.3**^{*3}	15/15
1: Cau	18	17	6	5.3	5.6	5.9	15/15
$\mathbf{f_{12}}$	110	270	370	460	1300	1500	15/15
0: BIP	**11**^{*3}	**7.4**^{*3}	**7.4**^{*3}	**7.7**^{*3}	**3.3**^{*3}	**3.3**^{*3}	15/15
1: Cau	79	41	35	38	17	17	15/15
$\mathbf{f_{13}}$	130	190	250	1300	1800	2300	15/15
0: BIP	**3.9**^{*3}	**5.4**^{*3}	**5.9**^{*3}	**1.6**^{*3}	**1.5**^{*3}	**1.7**^{*3}	15/15
1: Cau	21	24	25	7.4	7.3	7.3	15/15
$\mathbf{f_{14}}$	9.8	41	58	140	250	480	15/15
0: BIP	**1.1**^{*2}	**2.8**^{*3}	**3.7**^{*3}	**4.6**^{*3}	**5.4**^{*3}	**4.5**^{*3}	15/15
1: Cau	23	29	40	33	28	19	15/15
$\mathbf{f_{15}}$	510	9300	1.9e4	2.0e4	2.1e4	2.1e4	14/15
0: BIP	**1.6**^{*2}	**1.5**^{*2}	**1.2**^{*3}	**1.2**^{*3}	**1.2**^{*3}	**1.2**^{*3}	15/15
1: Cau	12	190	∞	∞	∞	∞ *2.5e5*	0/15
$\mathbf{f_{16}}$	120	610	2700	1.0e4	1.2e4	1.2e4	15/15
0: BIP	3	**3.6**^{*3}	**2.6**^{*3}	**1.3**^{*3}	**1.4**^{*3}	**1.4**^{*3}	15/15
1: Cau	5.6	1.2e3	∞	∞	∞	∞ *2.5e5*	0/15
$\mathbf{f_{17}}$	5.2	210	900	3700	6400	7900	15/15
0: BIP	3.4	**1**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	**1.2**^{*3}	15/15
1: Cau	44	13	7	4.3	5.3	13	14/15
$\mathbf{f_{18}}$	100	380	4000	9300	1.1e4	1.2e4	15/15
0: BIP	**1**^{*3}	**3.4**^{*2}	**1***	**1**^{*3}	**1.2**^{*3}	**1.3**^{*3}	15/15
1: Cau	13	12	2.4	2.7	3.7	8.6	14/15
$\mathbf{f_{19}}$	1	1	240	1.2e5	1.2e5	1.2e5	15/15
0: BIP	20	2.8e3	**160**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	15/15
1: Cau	300	2.1e4	∞	∞	∞	∞ *2.5e5*	0/15
$\mathbf{f_{20}}$	16	850	3.8e4	5.4e4	5.5e4	5.5e4	14/15
0: BIP	**3.3**^{*3}	**8.2**^{*3}	**2.8**^{*3}	**2.1**^{*3}	**2.2**^{*3}	**2.2**^{*3}	15/15
1: Cau	48	460	∞	∞	∞	∞ *2.5e5*	0/15
$\mathbf{f_{21}}$	41	1200	1700	1700	1700	1800	14/15
0: BIP	**2.3**^{*2}	14	24	25	25	25	15/15
1: Cau	20	27	190	420	420	410	4/15
$\mathbf{f_{22}}$	71	390	940	1000	1000	1100	14/15
0: BIP	6.9	20	45	**42**^{*2}	**41**^{*2}	**40**^{*2}	15/15
1: Cau	11	280	780	3.5e3	3.4e3	3.3e3	1/15
$\mathbf{f_{23}}$	3	520	1.4e4	3.2e4	3.3e4	3.4e4	15/15
0: BIP	1.7	**13***	**3.7**^{*3}	**1.8**^{*3}	**1.8**^{*3}	**1.8**^{*3}	15/15
1: Cau	2.2	230	∞	∞	∞	∞ *2.5e5*	0/15
$\mathbf{f_{24}}$	1600	2.2e5	6.4e6	9.6e6	1.3e7	1.3e7	3/15
0: BIP	**2.1**^{*3}	**1.6**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	3/15
1: Cau	30	∞	∞	∞	∞	∞ *2.5e5*	0/15

20-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
$\mathbf{f_1}$	43	43	43	43	43	43	15/15
0: BIP	**7.9**^{*3}	**14**^{*3}	**20**^{*3}	**33**^{*3}	**45**^{*3}	**57**^{*3}	15/15
1: Cau	730	1.6e3	2.5e3	4.3e3	6.1e3	7.8e3	15/15
$\mathbf{f_2}$	380	390	390	390	390	390	15/15
0: BIP	**35**^{*3}	**40**^{*3}	**44**^{*3}	**47**^{*3}	**48**^{*3}	**50**^{*3}	15/15
1: Cau	410	510	610	800	990	1.2e3	15/15
$\mathbf{f_3}$	5100	7600	7600	7600	7600	7700	15/15
0: BIP	**12**^{*3}	∞	∞	∞	∞	∞ *5.7e6*	0/15
1: Cau	∞	∞	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_4}$	4700	7600	7700	7700	7800	1.4e5	9/15
0: BIP	∞	∞	∞	∞	∞	∞ *5.5e6*	0/15
1: Cau	∞	∞	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_5}$	41	41	41	41	41	41	15/15
0: BIP	**5.1**^{*3}	**6.2**^{*3}	**6.3**^{*3}	**6.3**^{*3}	**6.3**^{*3}	**6.3**^{*3}	15/15
1: Cau	160	170	170	170	170	170	15/15
$\mathbf{f_6}$	1300	2300	3400	5200	6700	8400	15/15
0: BIP	**1.5**^{*3}	**1.3**^{*3}	**1.2**^{*3}	**1.1**^{*3}	**1.2**^{*3}	**1.2**^{*3}	15/15
1: Cau	1.0e3	1.3e3	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_7}$	1400	4300	9500	1.7e4	1.7e4	1.7e4	15/15
0: BIP	**1**^{*3}	**4.9**^{*3}	**3.5**^{*3}	**2.2**^{*3}	**2.2**^{*3}	**2.1**^{*3}	15/15
1: Cau	44	29	18	14	14	14	15/15
$\mathbf{f_8}$	2000	3900	4000	4200	4400	4500	15/15
0: BIP	**4**^{*3}	**4**^{*3}	**4.3**^{*3}	**4.5**^{*3}	**4.6**^{*3}	**4.6**^{*3}	15/15
1: Cau	190	180	210	260	360	540	4/15
$\mathbf{f_9}$	1700	3100	3300	3500	3600	3700	15/15
0: BIP	**4.7**^{*3}	**5.7**^{*3}	**6**^{*3}	**6.1**^{*3}	**6.1**^{*3}	**6.1**^{*3}	15/15
1: Cau	190	270	290	310	470	630	6/15
$\mathbf{f_{10}}$	7400	8700	1.1e4	1.5e4	1.7e4	1.7e4	15/15
0: BIP	**1.9**^{*3}	**1.8**^{*3}	**1.6**^{*3}	**1.2**^{*3}	**1.1**^{*3}	**1.1**^{*3}	15/15
1: Cau	20	22	20	20	21	25	15/15
$\mathbf{f_{11}}$	1000	2200	6300	9800	1.2e4	1.5e4	15/15
0: BIP	**10**^{*3}	**5.1**^{*3}	**1.9**^{*3}	**1.4**^{*3}	**1.2**^{*3}	**1**^{*3}	15/15
1: Cau	64	44	22	22	25	26	15/15
$\mathbf{f_{12}}$	1000	1900	2700	4100	1.2e4	1.4e4	15/15
0: BIP	**3**^{*3}	**4**^{*3}	**4.5**^{*3}	**4.5**^{*3}	**1.9**^{*3}	**2**^{*3}	15/15
1: Cau	510	440	420	380	390	1.1e3	0/15
$\mathbf{f_{13}}$	650	2000	2800	1.9e4	2.4e4	3.0e4	15/15
0: BIP	**4.3**^{*3}	**2.7**^{*3}	**5.1**^{*3}	**1.5**^{*3}	**2.3**^{*3}	**3**^{*3}	15/15
1: Cau	210	100	100	23	23	23	15/15
$\mathbf{f_{14}}$	75	240	300	930	1600	1.6e4	15/15
0: BIP	**3.9**^{*3}	**2.9**^{*3}	**3.7**^{*3}	**4.1**^{*3}	**6.2**^{*3}	**1.2**^{*3}	15/15
1: Cau	280	270	350	210	180	25	15/15
$\mathbf{f_{15}}$	3.0e4	1.5e5	3.1e5	3.2e5	4.5e5	4.6e5	15/15
0: BIP	**1**^{*3}	**2**^{*3}	**1.4**^{*3}	**1.4**^{*3}	**1**^{*3}	**1**^{*3}	15/15
1: Cau	∞	∞	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_{16}}$	1400	2.7e4	7.7e4	1.9e5	2.0e5	2.2e5	15/15
0: BIP	**1.7**^{*3}	**1**^{*3}	**1.2**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	15/15
1: Cau	∞	∞	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_{17}}$	63	1000	4000	3.1e4	5.6e4	8.0e4	15/15
0: BIP	**2.2**^{*3}	**1**^{*3}	**1**^{*3}	**1.2**^{*3}	**1.3**^{*3}	**1.4**^{*3}	15/15
1: Cau	260	120	62	16	23	∞ *1.0e6*	0/15
$\mathbf{f_{18}}$	620	4000	2.0e4	6.8e4	1.3e5	1.5e5	15/15
0: BIP	**1**^{*3}	**2.4**^{*3}	**1.2**^{*3}	**1.1**^{*3}	**1.7**^{*3}	**1.6**^{*3}	15/15
1: Cau	96	42	15	12	38	∞ *1.0e6*	0/15
$\mathbf{f_{19}}$	1	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
0: BIP	**170**^{*3}	**2.4e4**^{*3}	**1.2**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	15/15
1: Cau	8.4e3	∞	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_{20}}$	82	4.6e4	3.1e6	5.5e6	5.6e6	5.6e6	14/15
0: BIP	**4.3**^{*3}	**9.2**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	14/15
1: Cau	340	∞	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_{21}}$	560	6500	1.4e4	1.5e4	1.6e4	1.8e4	15/15
0: BIP	**3.2**^{*3}	**55**^{*3}	**48**^{*3}	**46**^{*3}	**43**^{*3}	**39**^{*3}	13/15
1: Cau	1.0e3	∞	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_{22}}$	470	5600	2.3e4	2.5e4	2.7e4	1.3e5	12/15
0: BIP	**6.8**^{*3}	**13**^{*3}	**210**^{*2}	**200**^{*2}	**190**^{*2}	**37**^{*2}	5/15
1: Cau	470	1.2e3	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_{23}}$	3.2	1600	6.7e4	4.9e5	8.1e5	8.4e5	15/15
0: BIP	4.3	**32**^{*3}	**1**^{*3}	**2**^{*3}	**1.2**^{*3}	**1.2**^{*3}	15/15
1: Cau	1.9	∞	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_{24}}$	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
0: BIP	**1**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	**1**^{*3}	3/15
1: Cau	∞	∞	∞	∞	∞	∞ *1.0e6*	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: BIP is BIPOP-CMA-ES and 1: Cau is CauchyEDA. 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.