Benchmarking a Weighted Negative Covariance Matrix Update on the BBOB-2010 Noisy Testbed

Nikolaus Hansen
INRIA Saclay, TAO project team
LRI, Bat 490 Univ. Paris-Sud
91405 Orsay Cedex, France
Nikolaus.Hansen@inria.fr

Raymond Ros
TAO Team-Project – INRIA Saclay
LRI, Bat 490, Univ. Paris-Sud
F-91405 Orsay Cedex, France
raymond.ros@inria.fr

ABSTRACT

In a companion paper, we presented a *weighted* negative update of the covariance matrix in the CMA-ES—weighted active CMA-ES or, in short, aCMA-ES. In this paper, we benchmark the IPOP-aCMA-ES on the BBOB-2010 noisy testbed in search space dimension between 2 and 40 and compare its performance with the IPOP-CMA-ES.

The aCMA suffers from a moderate performance loss, of less than a factor of two, on the sphere function with two different noise models. On the other hand, the aCMA enjoys a (significant) performance gain, up to a factor of four, on 13 unimodal functions in various dimensions, in particular the larger ones. Compared to the best performance observed during BBOB-2009, the IPOP-aCMA-ES sets a new record on overall ten functions. The global picture is in favor of aCMA which might establish a new standard also for noisy problems.

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, performance, comparison

Keywords

CMA-ES, IPOP-CMA-ES, active CMA-ES, Benchmarking, Black-box optimization

1. INTRODUCTION

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [9, 8, 7] is a stochastic search procedure that samples new candidate solutions from a multivariate normal distribution thereof mean and covariance matrix are adapted after each iteration. The $(\mu/\mu_\mathrm{w}, \lambda)$-CMA-ES samples λ new candidate solutions and selects the μ best among them. They contribute in a weighted manner to the update of the distribution parameters. The algorithm is non-elitist by nature, but a practical implementation will preserve the best-ever evaluated solution. Elitist variants of the CMA-ES [11] are often slightly faster but more susceptible of getting stuck in a suboptimal local optimum and far less attractive for noisy problems.

The IPOP-CMA-ES [1] implements a restart procedure. Before each restart, the population size λ is doubled. In most cases, doubling λ increases the length of a single run, but it often improves the quality of the best found solution in particular in a noisy environment. The BIPOP-CMA-ES, proposed recently [3], maintains two budgets. Under the first budget, an IPOP-CMA-ES is executed. Under the second budget, a multi-start $(\mu/\mu_\mathrm{w}, \lambda)$-CMA-ES with various small population sizes is entertained. The BIPOP-CMA-ES has performed exceptionally well on the BBOB-2009 noisy testbed [4][1], mainly thanks to the IPOP component.

A Further Improvement. The so-called active CMA-ES proposed in [12] introduces a negative update of the covariance matrix in the $(\mu/\mu_\mathrm{I}, \lambda)$-CMA-ES. In order to evaluate its performance, the authors investigate essentially unimodal functions. They observe a significant speed-up in particular on the discus function, because the negative update can in particular speed-up the adaptation of small variances in a small number of directions. The speed-up reaches almost a factor of three in dimension 20 (compared to the $(\mu/\mu_\mathrm{I}, \lambda)$-CMA-ES) and it increases with increasing dimension. This means that the update leads to an improved scaling with the search space dimension. The speed-up is less pronounced with a larger population size λ.

In the companion paper [10], the negative update of the covariance matrix has been implemented in a weighted fashion for the $(\mu/\mu_\mathrm{w}, \lambda)$-CMA-ES, denoted as $(\mu/\mu_\mathrm{w}, \lambda)$-aCMA-ES, in short aCMA in the following. On the noiseless BBOB-2010 testbed the aCMA dominated CMA, in that it was never significantly worse, but showed the expected improvement on ill-conditioned functions with a speed-up by a factor of up to three.

Objective of This Paper. In this paper we explore the IPOP-aCMA-ES [10] on noisy functions. Our main objectives are twofold. 1) search for possible flaws in the performance on a comprehensive noisy testbed. 2) quantify the

[1]See `http://coco.gforge.inria.fr/doku.php?id=bbob-2010-results`

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

performance advantage (as we expect an advantage) from aCMA compared to CMA. The algorithm is comprehensively described in [10].

2. METHODS

The experimental procedure is applied according to [5] on the 30 noisy benchmark functions given in [2, 6] for IPOP-aCMA-ES and IPOP-CMA-ES as presented in [10]. The

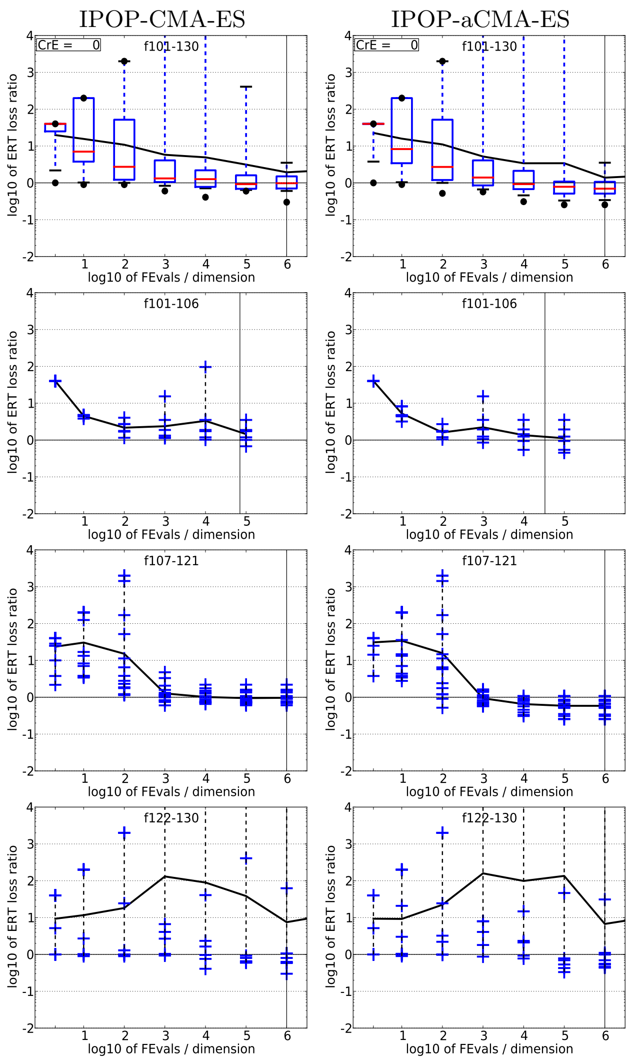

Figure 1: ERT ratio to the respective best algorithm from BBOB-2009 in dimension 20 versus a given budget FEvals**. The target value f_t for ERT is the smallest (best) recorded function value such that** $\mathrm{ERT}(f_t) \leq \mathrm{FEvals}$ **for the presented algorithm. Shown is** FEvals **divided by the respective best** $\mathrm{ERT}(f_t)$ **from BBOB-2009 for functions f_{101}–f_{130} in 5-D and 20-D. From top to bottom: for all functions, for moderate noisy functions, for unimodal and multi-modal functions. Line: geometric mean. Box-Whisker error bar: 25-75%-ile with median (box), 10-90%-ile (caps), and minimum and maximum** ERT **loss ratio (points). The vertical line gives the maximal number of function evaluations in this function subset.**

crafting-effort [5] of both algorithms is $\mathrm{CrE} = 0$. Measured CPU times are given in [10].

Source code to reproduce the experiment is provided at[2].

3. RESULTS

Runtime results comparing aCMA with CMA and with the respective best algorithm from BBOB-2009 are presented in Figures 1–4 and in Table 1. The **expected running time (ERT)**, used in the figures and table, depends on a given target function value, $f_t = f_{\mathrm{opt}} + \Delta f$, 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, 13]. **Statistical significance** is tested with the rank-sum test for a given target Δf_t (10^{-8} in Figure 4) up to the smallest number of function evaluations, $b^{\mathrm{u}}_{\mathrm{min}}$, found in any unsuccessful trial under consideration. The datum used from each trial is either the best achieved Δf-value, or if f_t was reached within the budget $b^{\mathrm{u}}_{\mathrm{min}}$, the number of needed function evaluations to reach Δf_t (inverted and multiplied by -1),

In the following, when a performance difference is highlighted on an individual function, the difference is statistically significant.

Figure 1. The figure shows the ERT ratio compared to the respective best algorithm from BBOB-2009 for all functions and for three function subgroups. Both algorithms show overall a very similar characteristic. Typical for the restart CMA-ES, the results become more competitive with an increasing budget of FEvals and therefore increasing difficulty of the function. For the final results, the median of IPOP-CMA-ES levels out with the set of best algorithms from BBOB-2009. IPOP-aCMA-ES is in 75% at most marginally slower than the best algorithm from BBOB-2009! The geometric mean loss, on the other hand, remains above one for both algorithms. This is, in fact, due to f_{126}, where both algorithms fail to reach a target value that the best algorithm hit with the first evaluation in dimension 20 (see Table 1)!

In all subgroups, both algorithms outperform the respective best algorithm from BBOB-2009 at least in some cases. Both algorithms show the best performance on the unimodal functions with "severe" noise (third row). On these function, also the (geometric) average ERT of aCMA drops below the average of the respective best algorithms from 2009 for larger budgets. On the multi-modal functions (last row), the ERT ratio shows a wide dispersion, again in particular due to f_{126}.

Figure 2. The figure shows empirical cumulative distributions (a) of the runtime in number of function evaluations and (b) of the runtime ratio between the two algorithms aCMA/CMA. Both algorithms perform very similar while aCMA appears to be slightly faster in all subfigures. The advantage is most pronounced on the unimodal (severe noisy) functions in dimension 20, where the ERT ratio shows in 50% of the cases a speed-up by about a factor of two (shift to the left of the lower part).

Figure 3. The scatter plots in Fig. 3 visualize the ratio of expected runtime aCMA/CMA for each measurement on each function and each dimension. Most of the points are

[2] `http://coco.gforge.inria.fr/doku.php?id=bbob-2010-results`

located close to the diagonal indicating again the similar performance.

We conjecture an advantage of aCMA on functions 104–106 (larger dimension), 113, 115–119, 121, often only in larger dimension.

In contrary, the CMA appears to be faster on functions 107 and 108. We investigate the relevance of these observations in

Figure 4. The question of a significant performance difference can be pursued in Fig. 4 which plots the ERT ratio aCMA/CMA versus the target function value and annotates final statistical significance. In the beginning, up to a function value of 1, we see large fluctuations that level out when better function values are reached. Then, the figure confirms advantages of aCMA on functions 104–106, 112, 113, 115–121, and 124 as statistically significant with a speedup of up to a factor of four on functions 115 and 119. Also the advantage of CMA on the sphere functions 107 and 108 is statistically significant.

Table 1. The table finally presents the ERT numbers for dimension 5 and 20 divided by the ERT of the respective best algorithm of BBOB-2009. We can see now that aCMA is significantly better than CMA on six functions in dimension 5 and on nine functions in dimension 20 (all from the above mentioned functions), where the final speed-up is more than a factor of 1.5 in all cases. Also the advantage of CMA on functions 107 and 108 shows up as significant in dimension 20.

Compared to the respectively best algorithm from BBOB-2009, the aCMA sets a new record in three and eight cases in dimension 5 and 20, respectively. In two more cases in dimension 5, the record is only significant for target value 10^{-5}. In some of these cases, also CMA improves significantly over the best algorithm from BBOB-2009, indicating that the IPOP component is the presumable main reason for the advantage. Function f_{126} was solved in dimension 5 for the first time with IPOP-aCMA-ES even though this could not be established as a statistically significant performance improvement.

Function 107. We pursued the question, why aCMA takes longer on function 107. The explanation meats our expectation: IPOP-aCMA-ES needs on average more restarts than IPOP-CMA-ES. In dimension 40, the final population size, which suffices to approach the optimum up to a target precision of 10^{-8}, is usually $\lambda = 240$ in IPOP-aCMA-ES, while it is $\lambda = 120$ in IPOP-CMA-ES. Presumably, the population size 120 is just below the necessary population size for aCMA-ES, while it just suffices for CMA-ES.

We imply from our observation that aCMA-ES might generally need a slightly larger population size on noisy problems.

4. SUMMARY AND CONCLUSION

The IPOP-aCMA-ES is a restarted $(\mu/\mu_{\mathrm{w}}, \lambda)$-CMA-ES with additional weighted negative update of the covariance matrix, as presented in the companion paper [10] and based on the key idea from [12]. In this paper we have evaluated IPOP-aCMA-ES on the BBOB-2010 noisy function testbed and compared with IPOP-CMA-ES. On the downside, we could observe a moderate performance decline on the sphere function. No (other) serious performance decline was detected. On the positive side, we found a significant speed-up of up to a factor of four with aCMA on 13 out of 30 functions in various dimensions, and in particular for the larger ones. The data leave not much doubt that the weighted negative covariance matrix update in aCMA is also an improvement for noisy functions, at least up to dimension 20.

Compared to the best performance seen in BBOB-2009, IPOP-aCMA-ES could set a new record on overall ten functions (considering dimension 5 and 20). Most statistically significant improvements were observe on unimodal functions with all three noise models. For the most severe noise model the improvement was less pronounced.

Overall, the additional weighted negative covariance matrix update in the IPOP-aCMA-ES has shown to be (a) reliable and (b) a relevant improvement over the original version also on noisy functions.

Acknowledgments

NH likes to thank the BBOB team for their excellent work and M. Schoenauer for his support of the BBOB project and beyond.

5. REFERENCES

[1] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 1769–1776. IEEE Press, 2005.

[2] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2010: Presentation of the noisy functions. Technical Report 2009/21, Research Center PPE, 2010.

[3] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In Rothlauf [14], pages 2389–2396.

[4] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 noisy testbed. In Rothlauf [14], pages 2397–2402.

[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: Noisy functions definitions. Technical Report RR-6869, INRIA, 2009. Updated February 2010.

[7] N. Hansen and S. Kern. Evaluating the CMA evolution strategy on multimodal test functions. In X. Yao et al., editors, *Parallel Problem Solving from Nature PPSN VIII*, volume 3242 of *LNCS*, pages 282–291. Springer, 2004.

[8] N. Hansen, S. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

[9] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[10] N. Hansen and R. Ros. Benchmarking a weighted negative covariance matrix update on the BBOB-2010

5-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
$\mathbf{f_{101}}$	11	37	44	62	69	75	15/15
0: CMA	3.3	3.4	4.7	6.0	7.8	9.3	15/15
1: aCMA	3.0	3.0	4.2	5.8	7.6	9.3	15/15
$\mathbf{f_{102}}$	11	35	50	72	86	99	15/15
0: CMA	3.4	3.1	4.1	5.1	6.5	7.3	15/15
1: aCMA	3.5	3.4	3.8	5.0	6.1	7.0	15/15
$\mathbf{f_{103}}$	11	28	30	31	35	115	15/15
0: CMA	3.6	4.0	6.6	12	17	7.1	15/15
1: aCMA	2.9	4.1	6.3	11	16	6.5	15/15
$\mathbf{f_{104}}$	173	773	1287	1768	2040	2284	15/15
0: CMA	1.4	3.4	2.9	2.7	2.5	2.4	15/15
1: aCMA	1.5	2.0	1.7	**1.6***	**1.5***	**1.4***	15/15
$\mathbf{f_{105}}$	167	1436	5174	10388	10824	11202	15/15
0: CMA	1.6	3.8	1.6	0.90	0.90	0.90	15/15
1: aCMA	2.8	3.0	1.3	0.70	0.70	0.70	15/15
$\mathbf{f_{106}}$	86	529	1050	2666	2887	3087	15/15
0: CMA	3.3	2.5	2.2	1.2	1.3	1.3	15/15
1: aCMA	2.5	1.9	1.6	**0.87***	**0.90***	**0.92**^{*2}	15/15
$\mathbf{f_{107}}$	40	228	453	940	1376	1850	15/15
0: CMA	2.1	0.98	1.1	1.3	1.2	1.1	15/15
1: aCMA	1.9	2.8	1.9	1.4	1.3	1.4	15/15
$\mathbf{f_{108}}$	87	5144	14469	30935	58628	80667	15/15
0: CMA	9.1	0.80	0.67	0.77	0.62	0.69	15/15
1: aCMA	11	0.68	0.64	0.64	0.66	0.94	15/15
$\mathbf{f_{109}}$	11	57	216	572	873	946	15/15
0: CMA	2.9	2.2	1.2	1.0	1.1	1.5	15/15
1: aCMA	3.1	1.9	1.1	0.92	1.2	1.5	15/15
$\mathbf{f_{110}}$	949	33625	1.20e5	5.93e5	6.03e5	6.11e5	15/15
0: CMA	0.73	8.3	3.4	0.72	0.73	0.74	15/15
1: aCMA	0.54	4.3	3.2	0.79	0.79	0.79	15/15
$\mathbf{f_{111}}$	6856	6.12e5	8.83e6	2.30e7	3.10e7	3.13e7	3/15
0: CMA	0.78	15	3.9	3.2	2.4	2.4	1/15
1: aCMA	0.54	7.3	7.9	∞	∞	∞ *5.0e6*	0/15
$\mathbf{f_{112}}$	107	1684	3421	4502	5132	5596	15/15
0: CMA	2.1	1.4	1.4	1.5	1.5	1.5	15/15
1: aCMA	2.2	1.2	**1.0***	**1.1**^{*2}	**1.1**^{*2}	**1.1**^{*2}	15/15
$\mathbf{f_{113}}$	133	1883	8081	24128	24128	24402	15/15
0: CMA	3.7	1.4	1.4	0.67	0.67	0.67	15/15
1: aCMA	4.4	0.78	0.77	0.32	0.32	0.32	15/15
$\mathbf{f_{114}}$	767	14720	56311	83272	83272	84949	15/15
0: CMA	3.2	0.45	0.48	0.79	0.79	0.80	15/15
1: aCMA	3.3	0.61	0.49	0.60	0.60	0.61	15/15
$\mathbf{f_{115}}$	64	485	1829	2550	2550	2970	15/15
0: CMA	1.7	2.4	2.7	3.1	3.1	2.7	15/15
1: aCMA	1.8	0.81	1.1	1.4	1.4	1.5	15/15
$\mathbf{f_{116}}$	5730	14472	22311	26868	30329	31661	15/15
0: CMA	3.1	2.3	1.9	1.8	1.7	1.7	15/15
1: aCMA	0.88	0.62	0.57	0.73	0.68	**0.68***	15/15
$\mathbf{f_{117}}$	26686	76052	1.10e5	1.37e5	1.73e5	1.92e5	15/15
0: CMA	1.1	0.95	0.77	0.73$^{\downarrow2}$	0.67$^{\downarrow2}$	0.69$^{\downarrow2}$	15/15
1: aCMA	0.65	0.48	0.44$^{\downarrow2}$	0.47$^{\downarrow2}$	0.43$^{\downarrow2}$	0.46$^{\downarrow2}$	15/15
$\mathbf{f_{118}}$	429	1217	1555	1998	2430	2913	15/15
0: CMA	3.2	2.0	1.9	2.0	1.9	1.7	15/15
1: aCMA	**1.7***	**0.88**^{*2}	**0.91***	**0.91**^{*3}	**0.91**^{*3}	**0.90**^{*3}	15/15
$\mathbf{f_{119}}$	12	657	1136	10372	35296	49747	15/15
0: CMA	1.1	0.35	0.70	0.83	1.0	1.4	15/15
1: aCMA	1.5	0.55	0.58	0.55	**0.49**$^{\star2\downarrow}$	**0.61**^{*2}	15/15
$\mathbf{f_{120}}$	16	2900	18698	72438	3.33e5	5.48e5	15/15
0: CMA	6.0	1.6	0.68	0.69	0.55$^{\downarrow2}$	0.83	15/15
1: aCMA	18	0.76	0.65	0.83	0.43$^{\downarrow2}$	0.66	15/15
$\mathbf{f_{121}}$	8.6	111	273	1583	3870	6195	15/15
0: CMA	1.9	1.1	1.0	1.1	2.1	2.3	15/15
1: aCMA	3.0	1.1	1.1	0.78	**0.84**^{*3}	**0.82**$^{\star3\downarrow}$	15/15
$\mathbf{f_{122}}$	10	1727	9190	30087	53743	1.11e5	15/15
0: CMA	4.8	0.94	0.44	0.56$^{\downarrow2}$	0.68	0.67$^{\downarrow}$	15/15
1: aCMA	2.9	1.4	0.71	0.48$^{\downarrow2}$	0.66	0.58$^{\downarrow}$	15/15
$\mathbf{f_{123}}$	11	16066	81505	3.36e5	6.71e5	2.22e6	15/15
0: CMA	23	0.62	0.52	0.74	0.65$^{\downarrow}$	0.45	15/15
1: aCMA	28	0.94	0.66	0.67	0.56$^{\downarrow}$	0.60	15/15
$\mathbf{f_{124}}$	10	202	1040	20478	45337	95200	15/15
0: CMA	2.8	1.3	4.0	1.2	0.93	0.65	15/15
1: aCMA	2.6	1.2	2.1	0.93	0.94	0.59	15/15
$\mathbf{f_{125}}$	1	1	1	2.39e5	2.43e5	2.46e5	15/15
0: CMA	1	27	2599	0.78	1.3	1.3	15/15
1: aCMA	1	26	3142	0.53	0.57	0.57	15/15
$\mathbf{f_{126}}$	1	1	1	∞	∞	∞	0
0: CMA	1	63	10254	*1.21e7*	*1.88e7*	*1.89e7*	2/15
1: aCMA	1	62	7882	∞	∞	∞	0/15
$\mathbf{f_{127}}$	1	1	1	3.42e5	3.89e5	3.95e5	15/15
0: CMA	1	15	1542	0.58	0.64	0.65	15/15
1: aCMA	1.1	30	3085	0.50	0.46	0.46	15/15
$\mathbf{f_{128}}$	111	4248	7808	12447	17217	21162	15/15
0: CMA	1.0	14	166	183	132	108	10/15
1: aCMA	2.6	1.8	47	30	22	18	14/15
$\mathbf{f_{129}}$	64	10710	59443	2.85e5	5.11e5	5.80e5	15/15
0: CMA	8.5	13	18	6.7	3.8	3.3	11/15
1: aCMA	11	10	16	7.0	3.9	3.5	10/15
$\mathbf{f_{130}}$	55	812	3034	32823	33889	34528	10/15
0: CMA	1.2	59	321	37	36	35	12/15
1: aCMA	1.4	143	391	36	35	35	11/15

20-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
$\mathbf{f_{101}}$	59	361	513	700	739	783	15/15
0: CMA	6.0	1.7	1.7	2.0	2.6	3.2	15/15
1: aCMA	6.1	1.8	1.7	2.1	2.7	3.3	15/15
$\mathbf{f_{102}}$	231	399	579	921	1157	1407	15/15
0: CMA	1.6	1.6	1.6	1.6	1.7	1.8	15/15
1: aCMA	1.5	1.6	1.6	1.6	1.7	1.8	15/15
$\mathbf{f_{103}}$	65	417	629	1313	1893	2464	14/15
0: CMA	5.5	1.5	1.4	1.2	1.2	1.2	15/15
1: aCMA	5.4	1.5	1.5	1.2	1.2	1.2	15/15
$\mathbf{f_{104}}$	23690	85656	1.71e5	1.82e5	1.89e5	1.96e5	15/15
0: CMA	7.5	2.5	1.3	1.3	1.3	1.2	15/15
1: aCMA	4.9	1.6	0.82	0.80	0.79	0.77	15/15
$\mathbf{f_{105}}$	1.92e5	6.11e5	6.32e5	6.49e5	6.60e5	6.70e5	15/15
0: CMA	1.9	0.76	0.76	0.77	0.77	0.76	15/15
1: aCMA	1.2	**0.43**$^{\star\downarrow3}$	**0.43**$^{\star\downarrow3}$	**0.44**$^{\star\downarrow3}$	**0.44**$^{\star\downarrow3}$	**0.44**$^{\star\downarrow3}$	15/15
$\mathbf{f_{106}}$	11480	21668	23746	25470	26492	27360	15/15
0: CMA	1.0	1.4	1.5	1.5	1.5	1.5	15/15
1: aCMA	**0.77***	**0.76**$^{\star3\downarrow3}$	**0.84**$^{\star3\downarrow2}$	**0.89**^{*3}	**0.91**^{*3}	**0.91**^{*3}	15/15
$\mathbf{f_{107}}$	8571	13582	16226	27357	52486	65052	15/15
0: CMA	1.1	0.95	1.1	0.96	**0.68***	**0.65**^{*2}	15/15
1: aCMA	0.81	0.97	1.3	1.5	1.3	1.1	15/15
$\mathbf{f_{108}}$	58063	97228	2.03e5	4.46e5	6.30e5	8.98e5	15/15
0: CMA	0.72	0.87	0.66	0.77	**0.94***	1.0	15/15
1: aCMA	0.74	0.98	1.0	1.1	1.5	1.5	15/15
$\mathbf{f_{109}}$	333	632	1138	2287	3583	4952	15/15
0: CMA	1.1	1.2	1.1	1.1	1.0	1.00	15/15
1: aCMA	1.1	1.2	1.2	1.2	1.1	1.1	15/15
$\mathbf{f_{110}}$	∞	∞	∞	∞	∞	∞	0
0: CMA	∞	∞	∞	∞	∞	∞	0/15
1: aCMA	∞	∞	∞	∞	∞	∞	0/15
$\mathbf{f_{111}}$	∞	∞	∞	∞	∞	∞	0
0: CMA	∞	∞	∞	∞	∞	∞	0/15
1: aCMA	∞	∞	∞	∞	∞	∞	0/15
$\mathbf{f_{112}}$	25552	64124	69621	73557	76137	78238	15/15
0: CMA	0.95	0.94	1.0	1.1	1.1	1.1	15/15
1: aCMA	0.84	0.84	0.89	0.93	0.94	0.93	15/15
$\mathbf{f_{113}}$	50123	3.64e5	5.60e5	5.88e5	5.88e5	5.91e5	15/15
0: CMA	1.0	0.53$^{\downarrow3}$	0.58$^{\downarrow3}$	0.59$^{\downarrow3}$	0.59$^{\downarrow3}$	0.59$^{\downarrow3}$	15/15
1: aCMA	0.60	0.27$^{\downarrow3}$	**0.27**$^{\star2\downarrow3}$	**0.29**$^{\star2\downarrow3}$	**0.29**$^{\star2\downarrow3}$	**0.29**$^{\star2\downarrow3}$	15/15
$\mathbf{f_{114}}$	2.08e5	1.12e6	1.45e6	1.57e6	1.57e6	1.58e6	15/15
0: CMA	0.59	0.68$^{\downarrow3}$	0.84	0.91	0.91	0.92	15/15
1: aCMA	0.64	0.35$^{\downarrow3}$	0.53$^{\downarrow}$	0.62	0.62	0.63	15/15
$\mathbf{f_{115}}$	2405	30268	91749	1.27e5	1.27e5	1.29e5	15/15
0: CMA	1.1	4.8	2.2	1.8	1.8	1.9	15/15
1: aCMA	1.4	**1.2**^{*3}	**0.54**^{*3}	**0.45**$^{\star3\downarrow3}$	**0.45**$^{\star3\downarrow3}$	**0.45**$^{\star3\downarrow3}$	15/15
$\mathbf{f_{116}}$	4.98e5	6.94e5	8.93e5	1.03e6	1.08e6	1.12e6	15/15
0: CMA	1.2	1.1	1.00	0.92	0.93	0.93	15/15
1: aCMA	**0.36**^{*2}	**0.28**^{*2}	**0.23**$^{\star2\downarrow}$	**0.22**$^{\star3\downarrow4}$	**0.23**$^{\star3\downarrow4}$	**0.24**$^{\star3\downarrow4}$	15/15
$\mathbf{f_{117}}$	1.79e6	2.46e6	2.60e6	2.91e6	3.24e6	3.62e6	15/15
0: CMA	0.55$^{\downarrow2}$	0.61$^{\downarrow4}$	0.66$^{\downarrow2}$	0.69$^{\downarrow2}$	0.71$^{\downarrow3}$	0.72$^{\downarrow2}$	15/15
1: aCMA	0.37$^{\downarrow2}$	0.35$^{\downarrow4}$	0.41$^{\downarrow2}$	0.45$^{\downarrow2}$	0.47$^{\downarrow3}$	0.56$^{\downarrow2}$	15/15
$\mathbf{f_{118}}$	6908	11786	17514	26342	30062	32659	15/15
0: CMA	2.0	1.8	1.7	1.5	1.5	1.5	15/15
1: aCMA	**0.82**^{*3}	**0.78**^{*3}	**0.71**^{*3}	**0.63**$^{\star3\downarrow3}$	**0.64**$^{\star3\downarrow3}$	**0.66**$^{\star3\downarrow3}$	15/15
$\mathbf{f_{119}}$	2771	29365	35930	4.11e5	1.40e6	1.90e6	15/15
0: CMA	1.9	0.58	0.69	0.58$^{\downarrow}$	0.59$^{\downarrow4}$	0.97	15/15
1: aCMA	1.6	0.62	0.83	0.42$^{\downarrow}$	**0.25**$^{\star2\downarrow4}$	**0.28**$^{\star3\downarrow3}$	15/15
$\mathbf{f_{120}}$	36040	1.79e5	2.81e5	1.59e6	6.74e6	1.35e7	13/15
0: CMA	0.69	0.60	0.74	0.67	0.69$^{\downarrow3}$	0.69$^{\downarrow2}$	15/15
1: aCMA	0.62	0.64	0.79	0.78	**0.44**$^{\star\downarrow3}$	**0.40**$^{\star\downarrow2}$	15/15
$\mathbf{f_{121}}$	249	769	1426	9304	34434	57404	15/15
0: CMA	1.3	1.1	1.1	1.1	1.4	1.9	15/15
1: aCMA	1.2	1.0	1.1	**0.77**$^{\star3\downarrow2}$	**0.56**$^{\star3\downarrow4}$	**0.64**$^{\star3\downarrow4}$	15/15
$\mathbf{f_{122}}$	692	52008	1.40e5	7.93e5	2.00e6	5.82e6	15/15
0: CMA	2.0	0.92	0.74	0.63	0.95	0.64	15/15
1: aCMA	2.3	0.81	0.97	0.56	0.66	0.81	15/15
$\mathbf{f_{123}}$	1063	5.30e5	1.49e6	5.29e6	2.71e7	1.58e8	0
0: CMA	7.2	0.72	0.61	0.80	0.62	∞ *2.0e7*	0/15
1: aCMA	6.4	0.72	0.88	0.94	0.50	1.9	0/15
$\mathbf{f_{124}}$	192	1959	40840	1.27e5	3.89e5	7.99e5	15/15
0: CMA	1.1	0.99	0.75	0.98	0.84	0.78	15/15
1: aCMA	1.1	3.9	1.0	0.91	0.76	0.62	15/15
$\mathbf{f_{125}}$	1	1	1	2.50e7	8.03e7	8.06e7	4/15
0: CMA	1	957	7.10e6	0.79	1.8	1.8	2/15
1: aCMA	1	827	3.85e6	1.1	1.7	1.7	2/15
$\mathbf{f_{126}}$	1	1	1	∞	∞	∞	0
0: CMA	1	5759	∞	∞	∞	∞	0/15
1: aCMA	1	6417	∞	∞	∞	∞	0/15
$\mathbf{f_{127}}$	1	1	1	4.43e6	7.27e6	7.43e6	15/15
0: CMA	1	267	9.58e5	0.81	0.84	0.85	15/15
1: aCMA	1	193	2.82e5	1.0	1.1	1.1	15/15
$\mathbf{f_{128}}$	1.40e5	1.34e7	1.72e7	1.72e7	1.72e7	1.72e7	9/15
0: CMA	12	1.0	1.4	1.4	1.4	1.4	6/15
1: aCMA	0.51	0.72	1.0	1.0	1.0	1.0	7/15
$\mathbf{f_{129}}$	7.81e6	4.13e7	4.15e7	4.18e7	4.21e7	4.24e7	5/15
0: CMA	0.29	0.30	0.77	0.77	0.77	0.77	4/15
1: aCMA	0.16	0.27	0.46	0.46	0.46	0.46	6/15
$\mathbf{f_{130}}$	4904	93149	2.52e5	2.54e5	2.55e5	2.57e5	7/15
0: CMA	4.9	76	37	37	37	37	9/15
1: aCMA	1.5	66	54	54	54	53	6/15

Table 1: 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_{101}–f_{130}. #succ is the number of trials that reached the final target $f_{\mathrm{opt}} + 10^{-8}$. Values in *italics* indicate number of function evaluations. 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, likewise the arrow indicates significance to the BBOB-2009 entry, all with Bonferroni correction of 60.

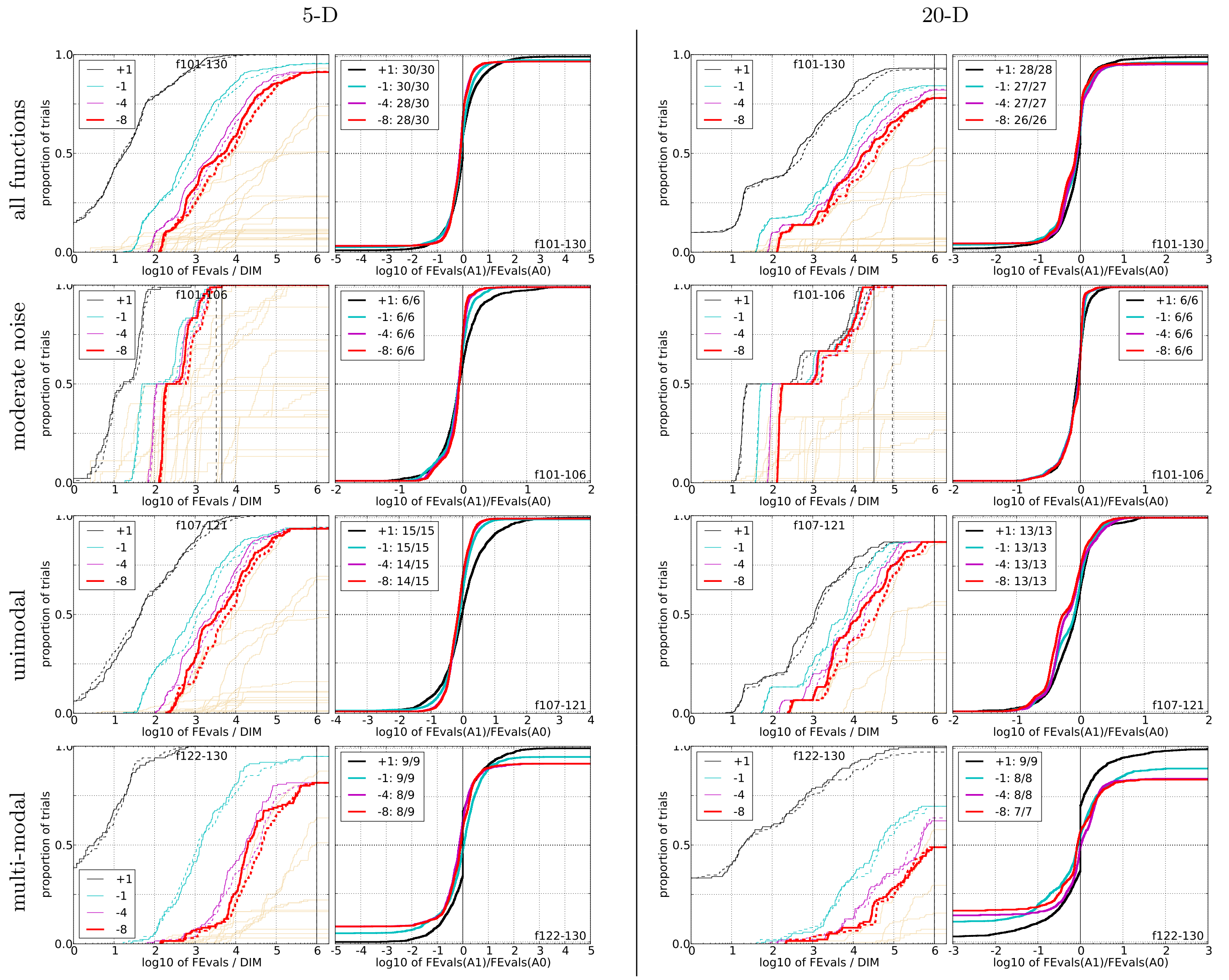

Figure 2: Empirical cumulative distributions (ECDF) of run lengths and speed-up ratios in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of necessary function evaluations divided by dimension D (FEvals/D) to reached a target value $f_{\rm opt} + \Delta f$ with $\Delta f = 10^k$, where $k \in \{1, -1, -4, -8\}$ is given by the first value in the legend, for IPOP-aCMA (solid) and IPOP-CMA (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 of IPOP-aCMA divided by IPOP-CMA, 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 (IPOP-aCMA first).

noiseless testbed. In J. Branke, editor, *GECCO (Companion)*. ACM, 2010.

[11] C. Igel, T. Suttorp, and N. Hansen. A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In *Proceedings of the 8th annual conference on genetic and evolutionary computation GECCO*, pages 453–460. ACM, 2006.

[12] G. Jastrebski and D. Arnold. Improving evolution strategies through active covariance matrix adaptation. In *IEEE Congress on Evolutionary Computation, CEC 2006, Proceedings*, pages 2814–2821. IEEE, 2006.

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

[14] F. Rothlauf, editor. *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009, Companion Material.* ACM, 2009.

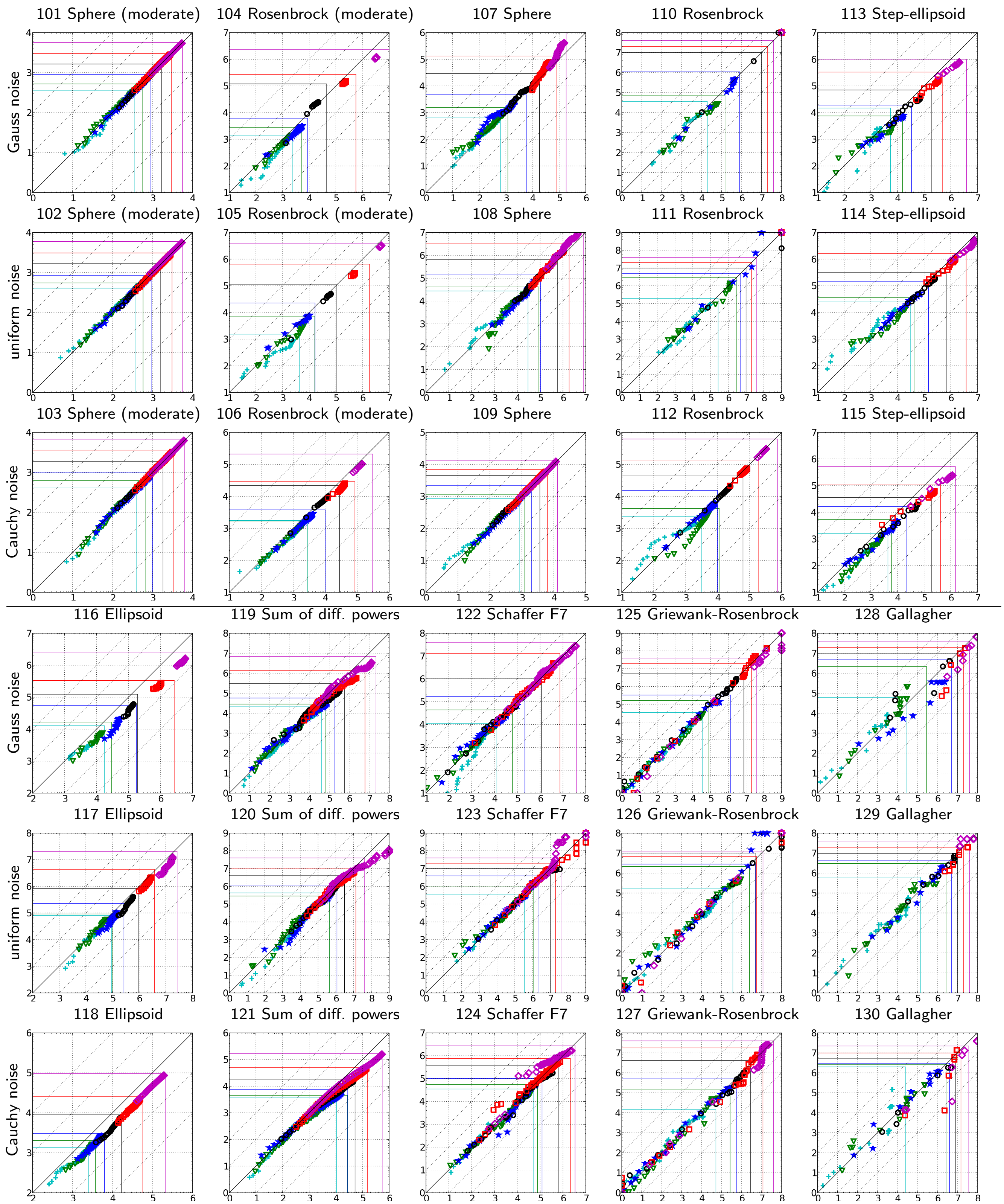

Figure 3: Expected running time (ERT in log10 of number of function evaluations) of IPOP-aCMA versus IPOP-CMA for 46 target values $\Delta f \in [10^{-8}, 10]$ in each dimension for functions f_{101}–f_{130}. Markers on the upper or right egde indicate that the target value was never reached by IPOP-aCMA or IPOP-CMA respectively. Markers represent dimension: 2:+, 3:▽, 5:⋆, 10:○, 20:□, 40:◇. Lines indicate the maximum number of function evaluations

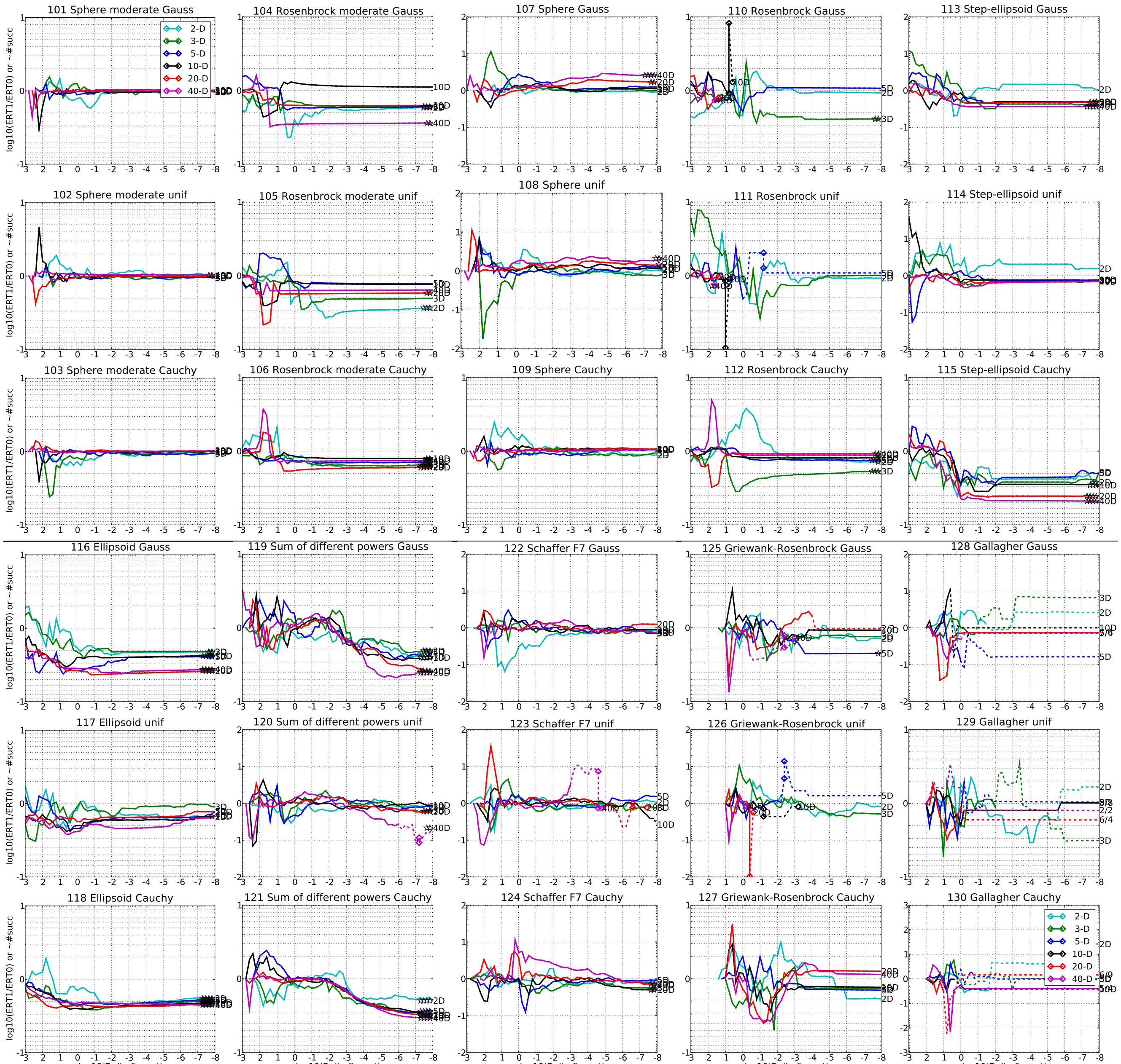

Figure 4: Ratio of the expected running times (ERT) of IPOP-aCMA divided by IPOP-CMA versus $\log_{10}(\Delta f)$ for f_{101}–f_{130} in 2, 3, 5, 10, 20, 40-D. Ratios $< 10^0$ indicate an advantage of IPOP-aCMA, smaller values are always better. The line gets dashed when for any algorithm the ERT exceeds 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 IPOP-aCMA. 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 IPOP-aCMA (1st number) and non-zero for IPOP-CMA (2nd number). Results are statistically significant with $p = 0.05$ for one star and $p = 10^{-\#\star}$ otherwise, with Bonferroni correction within each figure.