An ACO Algorithm Benchmarked on the BBOB Noiseless Function Testbed

Tianjun Liao
IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
tliao@ulb.ac.be

Daniel Molina
Dept. of Computer Engineering, University of Cadiz, Cadiz, Spain
daniel.molina@uca.es

Thomas Stützle
IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
stuetzle@ulb.ac.be

Marco A. Montes de Oca
Dept. of Mathematical Sciences, University of Delaware, Newark, DE, USA
mmontes@math.udel.edu

Marco Dorigo
IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
mdorigo@ulb.ac.be

ABSTRACT

$\text{ACO}_{\mathbb{R}}$ is an ant colony optimization algorithm for continuous domains. In this article, we benchmark $\text{ACO}_{\mathbb{R}}$ on the BBOB noiseless function testbed, and compare its performance to PSO, ABC and GA algorithms from previous BBOB workshops. Our experiment shows that $\text{ACO}_{\mathbb{R}}$ performs better than PSO, ABC and GA on the moderate functions, ill-conditioned functions and multi-modal functions. Among 24 functions, $\text{ACO}_{\mathbb{R}}$ solved 19 in dimension 5, 9 in dimension 20, and 7 across dimensions from 2 to 40. Furthermore, in dimension 5, we present the results of the $\text{ACO}_{\mathbb{R}}$ when it uses variable correlation handling. The latter version is competitive on the five dimensional functions to (1+1)-CMA-ES and BIPOP-CMA-ES.

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, Ant colony optimization, Continuous domains

1. INTRODUCTION

The ant colony optimization (ACO) metaheuristic was originally proposed for solving discrete optimization problems [2]. Recently, the adaption of ACO algorithms for continuous domains received increasing attention [9, 11, 13]. Socha and Dorigo [13] replaced the discrete probability distribution with probability density functions (PDFs) in the solution construction for continuous domains, and thus proposed an ACO algorithm for continuous domains, called $\text{ACO}_{\mathbb{R}}$. The popularity of $\text{ACO}_{\mathbb{R}}$ is illustrated by the more than 260 citations according to Google Scholar as of March 2012 and by being one of top 10 cited papers of the recent five years in the European Journal of Operational Research. However, $\text{ACO}_{\mathbb{R}}$ has not been benchmarked so far on the BBOB function testbed.

In this article, we benchmark $\text{ACO}_{\mathbb{R}}$ on the BBOB noiseless function testbed. We test two versions of $\text{ACO}_{\mathbb{R}}$. The first version uses the original mechanism, proposed in [13] to handle variable correlations; the second version does not use this mechanism. In what follows, these two versions are called $\text{ACO}_{\mathbb{R}}$-vch and $\text{ACO}_{\mathbb{R}}$, respectively. As a better illustration, we compare the performance of $\text{ACO}_{\mathbb{R}}$ to the data obtained by three standard nature-inspired algorithms PSO [4], ABC [3], and GA [12] which have been benchmarked in the previous BBOB workshops. Furthermore, we compare $\text{ACO}_{\mathbb{R}}$-vch to performance data for (1+1)-CMA-ES [1] and for BIPOP-CMA-ES [6] from the BBOB 2009 workshop.

2. ALGORITHM PRESENTATION

$\text{ACO}_{\mathbb{R}}$ [13] uses a solution archive to create a probability distribution of promising solutions over the search space. The solution archive is initialized by k random solutions. The algorithm iteratively updates the solution archive by generating m new solutions and then keeping only the best k solutions of the $k+m$ solutions. Solutions are generated variable by variable based on a Gaussian kernel, which is defined as a weighted sum of several Gaussian functions g_j^i, where j is a solution index and i is a variable index. The Gaussian kernel for variable i is:

$$G^i(x) = \sum_{j=1}^{k} \omega_j g_j^i(x) = \sum_{j=1}^{k} \omega_j \frac{1}{\sigma_j^i \sqrt{2\pi}} e^{-\frac{(x-\mu_j^i)^2}{2{\sigma_j^i}^2}} , \quad (1)$$

where $j \in \{1, ..., k\}$, $i \in \{1, ..., D\}$, with D being the problem dimensionality, and ω_j is a weight associated with the ranking of solution s_j in the archive, $rank(j)$. ω_j is defined by:

$$\omega_j = \frac{1}{qk\sqrt{2\pi}} e^{\frac{-(rank(j)-1)^2}{2q^2k^2}} , \quad (2)$$

where q is a parameter. In $g_j^i(x)$ of Equation 1, $\mu_j^i = s_j^i$, and σ_j^i is equal to

$$\sigma_j^i = \xi \sum_{r=1}^{k} \frac{|s_r^i - s_j^i|}{k-1} , \quad (3)$$

where ξ is a parameter. The $ACO_{\mathbb{R}}$ we test here is based on a re-implementation in C++ of the original implementation in R that was used in [13].

3. EXPERIMENTAL PROCEDURE

We use here the parameter values that were recommended in the original paper [13], that is: m=2, k=50, q=0.1, ξ=0.85. A maximum of 10^7 function evaluations was used. Every periodic 25000 iterations with a relative solution improvement less than 10^{-8}, $ACO_{\mathbb{R}}$ restarts without forgetting the best-so-far solution. To ensure that the final best solution is inside the bounds, the bound constraints are enforced by clamping each generated solution that violates the bound constraint to the nearest solution on the bounds. The negative impact of an infeasible final solution outside the bounds on algorithm comparisons was presented by Liao et al. [10].

4. RESULTS

Results from experiments following the procedure in [7] on the benchmark functions from [5, 8] are presented in Figures 1 2, and 3 and in Tables 1 and 2.

Among the 24 functions, $ACO_{\mathbb{R}}$ solved 19 (16 with a 100% success rate) in dimension 5 and 9 (6 with a 100% success rate) in dimension 20. $ACO_{\mathbb{R}}$ solved all the moderate and multi-modal functions in dimension 5, in which $ACO_{\mathbb{R}}$ almost reaches a 100% success rate for all these functions expect one failure trial in f_{19}. $ACO_{\mathbb{R}}$ solved f_1, f_2, f_5, f_6, f_8, f_9, f_{21} over dimensions from 2 to 40.

We compare the performance of $ACO_{\mathbb{R}}$ to the data obtained by PSO, ABC and GA in previous BBOB workshops. As seen from Figures 2 and 3, we observe that $ACO_{\mathbb{R}}$ obtains better performance than the references when comprehensively considering all functions. Figures 2 clearly illustrates that $ACO_{\mathbb{R}}$ obtains better run-time performance than PSO, ABC and GA on the moderate functions, ill-conditioned functions and multi-modal functions. Especially on the moderate functions, across dimensions 5 and 20, $ACO_{\mathbb{R}}$ clearly dominates PSO, ABC and GA.

We also observe that $ACO_{\mathbb{R}}$ solved two Rosenbrock functions (f_8 and f_9) on dimension 20 with a 100% success rate, and solved two Schaffers F7 functions (f_{17} and f_{18}) on dimension 5 with a 100% success rate. However, $ACO_{\mathbb{R}}$ does not perform very good on multi-modal functions of higher dimensions and even some weakly structured functions of lower dimensions. In the comparisons, PSO is the only one that could solve the Katsuura function (f_{23}) of dimension 2 and 3; ABC obtained the best performance on the two separable Rastrigin functions.

5. CPU TIMING EXPERIMENT

The $ACO_{\mathbb{R}}$ was run on f_8 until at least 30 seconds have passed. These experiment were conducted with Intel Xeon E5410 (2.33 GHz) on Linux (kernel 2.6.9 - 78.0.22). The results were 3.0E-06, 3.0E-06, 6.5E-04, 7.5E-04, 9.5E-04 and 1.4E-03 seconds per function evaluation in dimensions 2, 3, 5, 10, 20, and 40, respectively.

6. DISCUSSION

We additionally present some performance results of $ACO_{\mathbb{R}}$-vch comparing it to the data obtained by (1+1)-CMA-ES and BIPOP-CMA-ES in the BBOB 2009 workshop. We restrict the comparison to functions of 5 dimensions. In Figure 4, we observe that $ACO_{\mathbb{R}}$-vch greatly improves over $ACO_{\mathbb{R}}$ in functions with moderate or high conditioning ($f_6 - f_{14}$) and that $ACO_{\mathbb{R}}$-vch performs very competitive to (1+1)-CMA-ES and BIPOP-CMA-ES. In the separable, multi-modal and weakly structured functions, $ACO_{\mathbb{R}}$-vch performs slightly worse than $ACO_{\mathbb{R}}$, while $ACO_{\mathbb{R}}$-vch performs clearly better than $ACO_{\mathbb{R}}$ on moderate and ill-conditioned functions. Both $ACO_{\mathbb{R}}$ and $ACO_{\mathbb{R}}$-vch obtain a better performance than (1+1)-CMA-ES in the separable, multi-modal functions, or when comprehensively considering all functions. In the weakly structured functions and multi-modal functions they perform worse that BIPOP-CMA-ES, while they perform better on the separable functions.

7. CONCLUSION

In this article, we present benchmark results for a re-implementation of $ACO_{\mathbb{R}}$ on the BBOB noiseless function testbed. Furthermore, we discuss the performance of $ACO_{\mathbb{R}}$-vch with variable correlation handling. It is observed that the latter version is competitive to (1+1)-CMA-ES and BIPOP-CMA-ES in functions with moderate or high conditioning.

8. ACKNOWLEDGMENTS

The authors would like to thank the great and hard work of the BBOB team. This work was supported by the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n° 246939 and by the Meta-X project funded by the Scientific Research Directorate of the French Community of Belgium. Thomas Stützle and Marco Dorigo acknowledge support from the Belgian F.R.S.-FNRS, of which they are a Research Associate and a Research Director, respectively. Tianjun Liao acknowledges a fellowship from the China Scholarship Council.

9. REFERENCES

[1] A. Auger and N. Hansen. Benchmarking the (1+1)-CMA-ES on the BBOB-2009 function testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'09*, pages 2459–2466, New York, NY, USA, 2009. ACM.

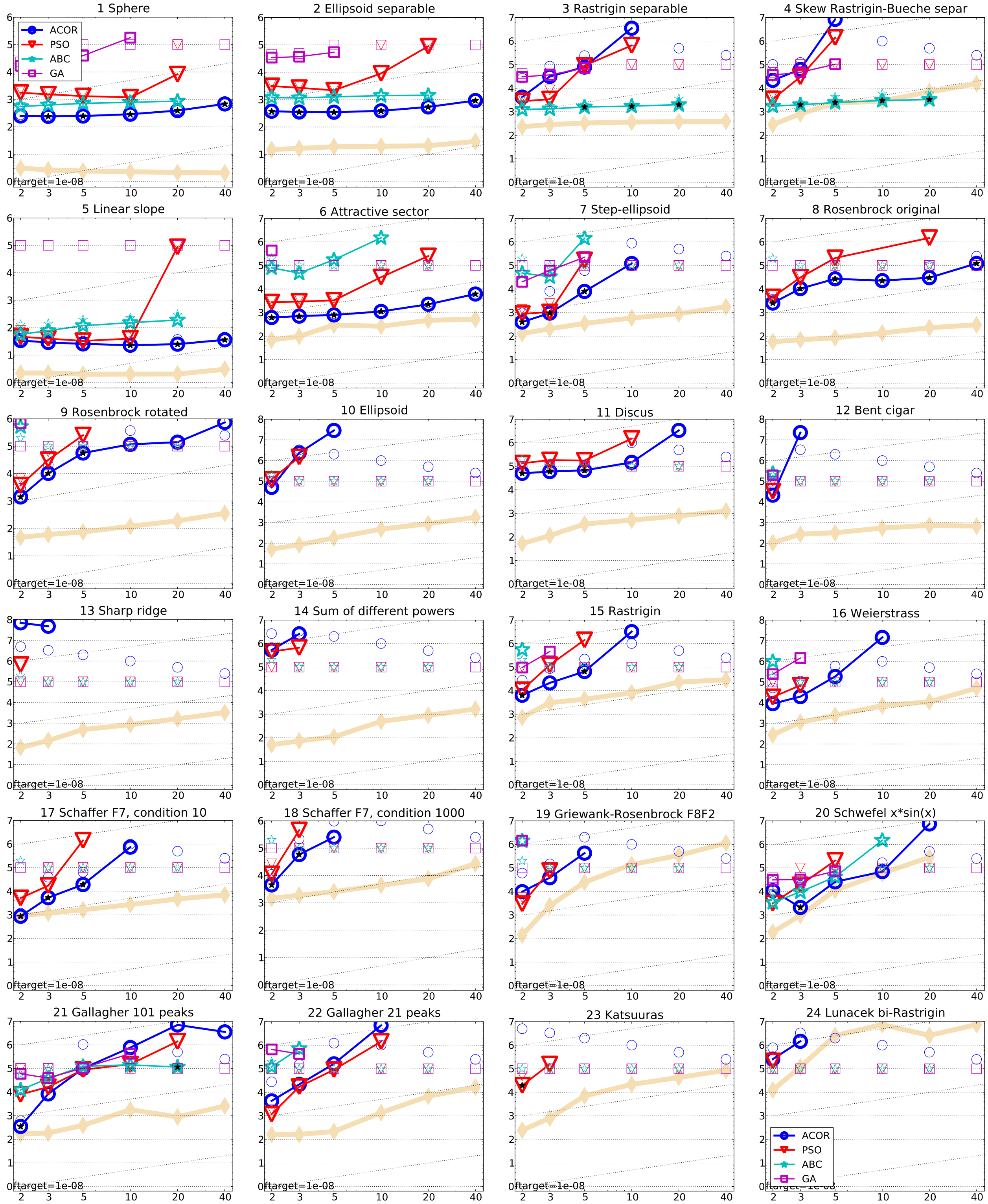

Figure 1: Expected running time (ERT in number of f-evaluations) divided by dimension for target function value 10^{-8} as $\log_{10}$ values versus dimension. Different symbols correspond to different algorithms given in the legend of f_1 and f_{24}. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Horizontal lines give linear scaling, slanted dotted lines give quadratic scaling. Black stars indicate statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: ∘:ACOR, ▽:PSO, ⋆:ABC, □:GA

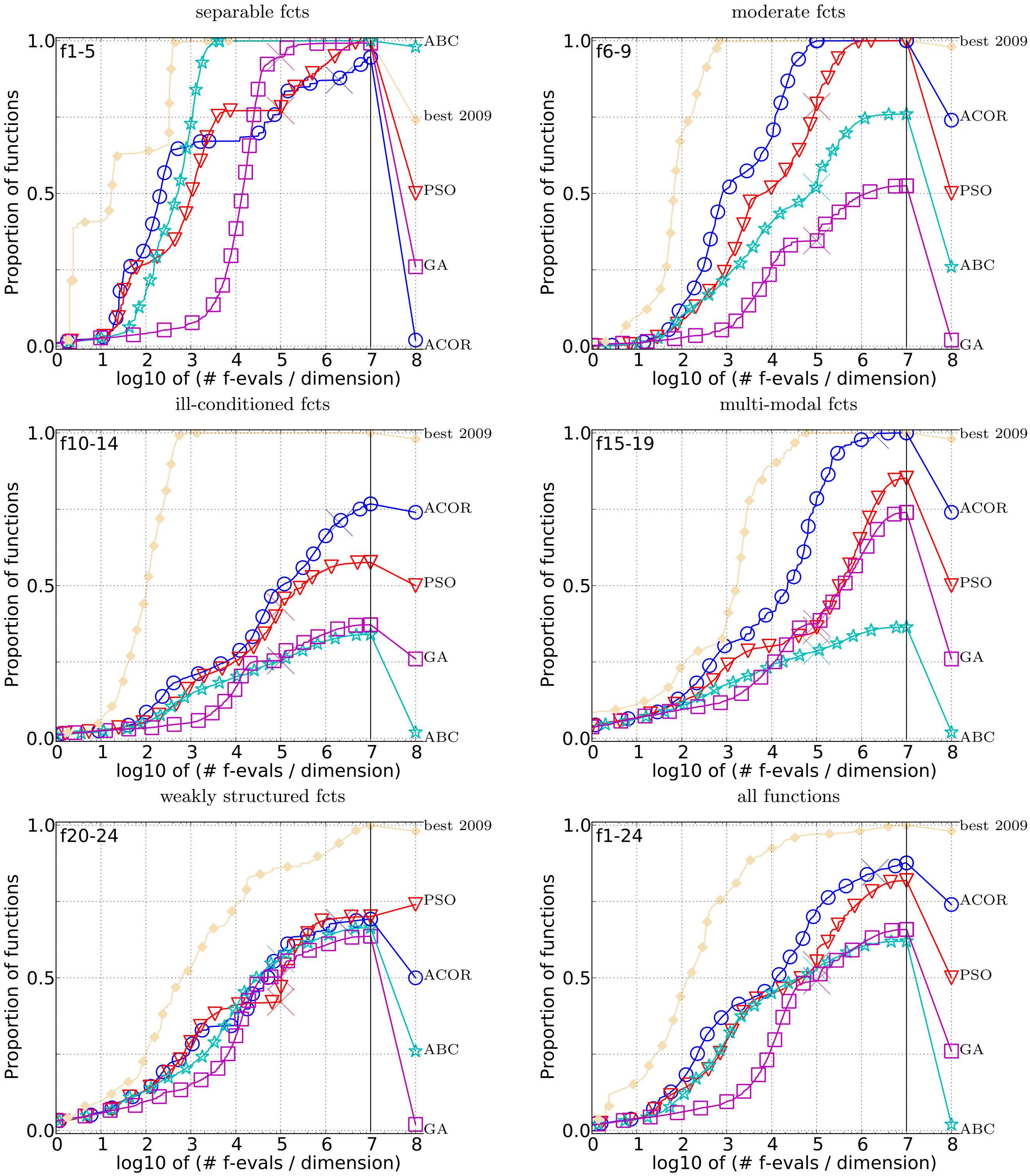

Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 5-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single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

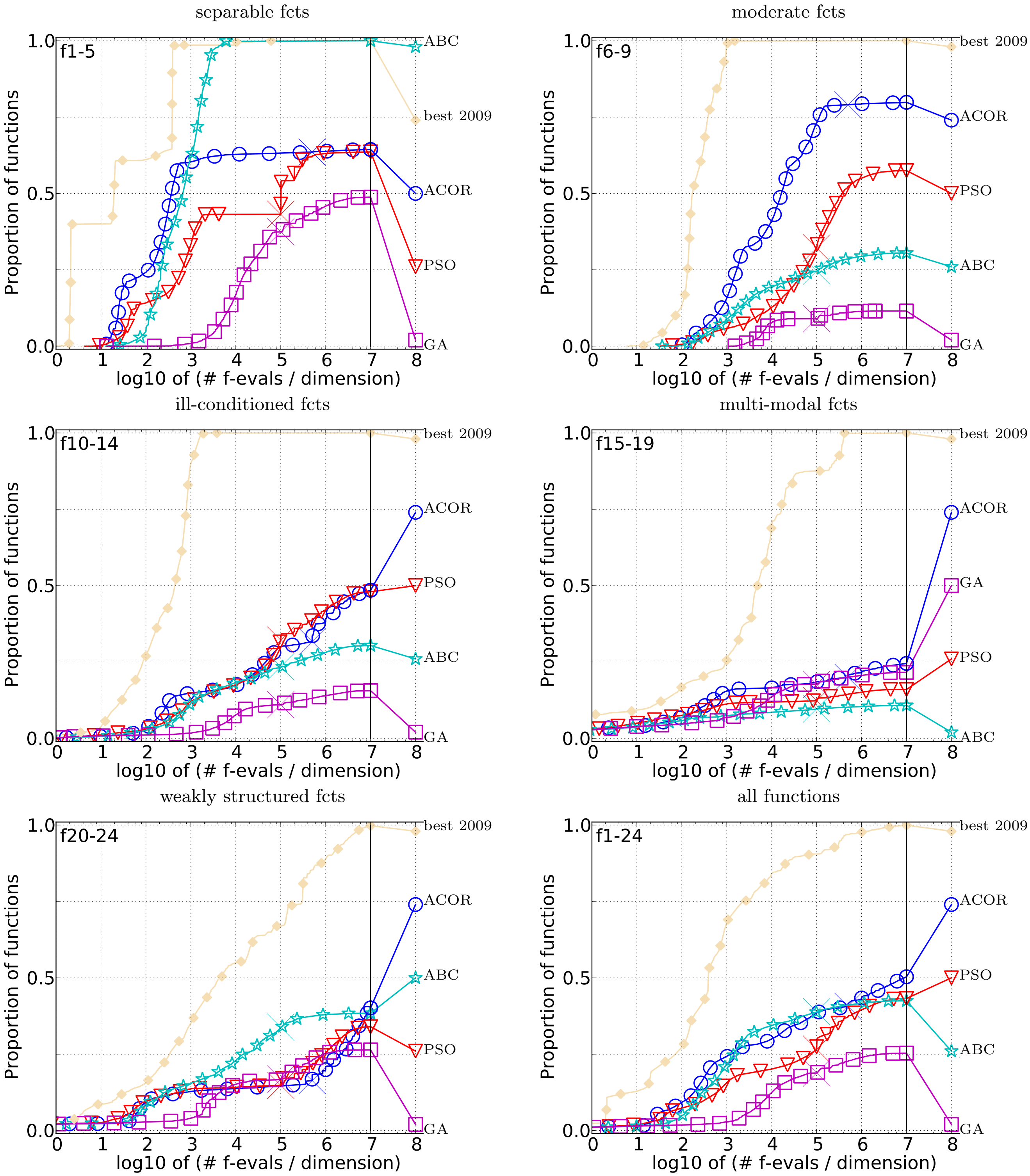

Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 20-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single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

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f1	11	12	12	12	12	12	15/15
ACOR	4.7(5)	**15**(3)*	**26**(4)$^{\star 2}$	**47**(6)$^{\star 4}$	**67**(4)$^{\star 4}$	**90**(6)$^{\star 4}$	15/15
PSO	**3.7**(3)	22(6)	55(18)	182(30)	317(45)	450(55)	15/15
ABC	12(14)	32(24)	62(30)	122(25)	191(14)	255(15)	15/15
GA	8.7(7)	362(196)	1182(175)	2940(390)	5384(387)	8329(789)	13/15
f2	83	87	88	90	92	94	15/15
ACOR	**6.0**(0.6)$^{\star 3}$	**7.4**(0.6)$^{\star 3}$	**8.7**(0.8)$^{\star 4}$	**12**(1)$^{\star 4}$	**14**(0.8)$^{\star 4}$	**17**(1)$^{\star 4}$	15/15
PSO	32(8)	41(6)	49(5)	68(8)	89(11)	105(11)	15/15
ABC	11(6)	18(11)	26(10)	38(13)	50(10)	62(7)	15/15
GA	333(63)	456(52)	606(74)	1304(66)	2158(2751)	2530(2707)	13/15
f3	716	1622	1637	1646	1650	1654	15/15
ACOR	1.7(0.8)	30(38)	241(167)	240(166)	239(165)	239(165)	15/15
PSO	52(2)	55(155)	275(458)	275(307)	276(307)	276(304)	8/15
ABC	**1.0**(0.6)	**1.5**(0.6)$^{\star 3}$	**1.8**(0.6)$^{\star 4}$	**2.7**(0.5)$^{\star 4}$	**3.6**(0.5)$^{\star 3}$	**4.4**(0.7)$^{\star 2}$	15/15
GA	19(2)	18(2)	25(2)	43(4)	112(156)	200(162)	11/15
f4	809	1633	1688	1817	1886	1903	15/15
ACOR	2.0(1)	783(1075)	2.5e4(3e4)	2.3e4(3e4)	2.2e4(2e4)	2.2e4(3e4)	3/15
PSO	3.0(1.0)	141(163)	4152(4813)	3859(4746)	3720(4110)	3687(4203)	1/15
ABC	**1.1**(0.6)*	**2.4**(1)$^{\star 3}$	**2.9**(1.0)$^{\star 4}$	**4.3**(2)$^{\star 3}$	**4.9**(2)$^{\star 4}$	**6.0**(2)$^{\star 4}$	15/15
GA	18(4)	20(2)	26(3)	41(7)	58(5)	185(141)	9/15
f5	10	10	10	10	10	10	15/15
ACOR	**8.5**(1)	**12**(2)	**13**(3)	**13**(3)	**13**(3)	**13**(3)	15/15
PSO	10(2)	14(5)	16(6)	16(6)	16(6)	16(6)	15/15
ABC	32(17)	49(28)	58(34)	59(35)	59(35)	59(35)	15/15
GA	481(267)	2072(463)	3983(388)	9220(765)	1.7e4(13833)	3.4e4(4852)	0/15
f6	114	214	281	580	1038	1332	15/15
ACOR	**3.4**(1)	**3.6**(0.8)$^{\star 3}$	**4.1**(0.9)$^{\star 4}$	**3.4**(0.4)$^{\star 4}$	**2.6**(0.4)$^{\star 4}$	**2.7**(0.3)$^{\star 4}$	15/15
PSO	4.7(2)	9.0(5)	11(4)	11(2)	10(2)	11(1)	15/15
ABC	4.9(3)	15(10)	365(891)	619(881)	498(722)	507(674)	6/15
GA	66(56)	148(60)	381(80)	1.2e4(1e4)	∞	∞ *5e5*	0/15
f7	24	324	1171	1572	1572	1597	15/15
ACOR	**6.5**(3)	**2.1**(1)	**32**(25)	**25**(18)$^{\star 2}$	**25**(18)$^{\star 2}$	**25**(18)$^{\star 2}$	15/15
PSO	11(7)	9.5(15)	587(807)	541(673)	541(578)	533(670)	6/15
ABC	20(29)	16(14)	62(58)	957(1230)	957(1113)	1359(1566)	1/15
GA	50(56)	35(14)	57(12)	524(640)	524(640)	523(630)	5/15
f8	73	273	336	391	410	422	15/15
ACOR	**5.5**(0.9)	**11**(2)	**43**(4)	**120**(11)$^{\star 2}$	**199**(10)$^{\star 4}$	**278**(10)$^{\star 4}$	15/15
PSO	13(4)	153(14)	201(76)	467(112)	781(138)	1103(132)	7/15
ABC	6.0(4)	12(10)	52(17)	2509(2946)	∞	∞ *5e5*	0/15
GA	186(35)	837(975)	∞	∞	∞	∞ *5e5*	0/15
f9	35	127	214	300	335	369	15/15
ACOR	**12**(3)	**21**(8)	**59**(26)	**251**(139)	**467**(266)	**655**(376)	15/15
PSO	25(13)	938(1971)	678(1197)	1129(1116)	2361(2609)	2753(2720)	5/15
ABC	14(9)	69(113)	699(808)	∞	∞	∞ *5e5*	0/15
GA	423(82)	5.6e4(7e4)	∞	∞	∞	∞ *5e5*	0/15
f10	349	500	574	626	829	880	15/15
ACOR	**662**(773)	**1848**(1729)	**3067**(1662)	**5253**(2231)	**5798**(2317)	**9645**(6851)	1/15
PSO	1739(2099)	3260(3445)	∞	∞	∞	∞ *5e5*	0/15
ABC	2.1e4(2e4)	∞	∞	∞	∞	∞ *5e5*	0/15
GA	2372(2927)	∞	∞	∞	∞	∞ *5e5*	0/15
f11	143	202	763	1177	1467	1673	15/15
ACOR	130(102)	258(72)	**118**(34)	**130**(26)	**153**(30)$^{\star 2}$	**177**(34)$^{\star 3}$	15/15
PSO	**91**(61)	**236**(121)	123(68)	164(65)	243(52)	391(303)	8/15
ABC	160(240)	6088(6581)	9336(1e4)	∞	∞	∞ *5e5*	0/15
GA	339(65)	7131(8722)	9353(1e4)	∞	∞	∞ *5e5*	0/15
f12	108	268	371	461	1303	1494	15/15
ACOR	4129(7120)	3.3e4(4e4)	7.8e4(9e4)	∞	∞	∞ *1e7*	0/15
PSO	747(2312)	3750(4669)	**5412**(6746)	**1.5e4**(2e4)	∞	∞ *5e5*	0/15
ABC	**99**(78)	**547**(474)	5852(6746)	∞	∞	∞ *5e5*	0/15
GA	931(148)	2449(2851)	1.9e4(2e4)	∞	∞	∞ *5e5*	0/15
f13	132	195	250	1310	1752	2255	15/15
ACOR	99(221)	475(603)	**2137**(2399)	**5448**(7655)	**8.1e4**(9e4)	∞ *1e7*	0/15
PSO	1579(1900)	1.0e4(1e4)	2.8e4(3e4)	∞	∞	∞ *5e5*	0/15
ABC	**18**(15)	**187**(186)	6618(7006)	∞	∞	∞ *5e5*	0/15
GA	242(30)	728(1316)	4394(5025)	∞	∞	∞ *5e5*	0/15
f14	10	41	58	139	251	476	15/15
ACOR	**1.7**(2)	**4.7**(2)	**6.3**(1)$^{\star 4}$	**9.3**(3)$^{\star 4}$	**123**(85)	**4.8e4**(5e4)	0/15
PSO	1.9(2)	5.6(3)	15(4)	30(8)	218(157)	∞ *5e5*	0/15
ABC	3.5(3)	11(8)	19(7)	679(802)	∞	∞ *5e5*	0/15
GA	2.1(2)	91(64)	267(66)	350(71)	∞	∞ *5e5*	0/15
f15	511	9310	19369	20073	20769	21359	14/15
ACOR	**5.3**(4)	**7.7**(11)*	**17**(27)$^{\star 3}$	**16**(26)$^{\star 3}$	**16**(25)$^{\star 3}$	**15**(25)$^{\star 3}$	15/15
PSO	16(7)	221(269)	366(394)	353(411)	342(391)	333(375)	1/15
ABC	15(7)	243(279)	∞	∞	∞	∞ *5e5*	0/15
GA	35(8)	91(108)	367(382)	355(431)	345(381)	∞ *5e5*	0/15
f16	120	612	2662	10449	11644	12095	15/15
ACOR	7.0(9)	325(232)	154(187)	**66**(77)	**70**(66)	**75**(83)	15/15
PSO	2.4(3)	**6.2**(5)	**59**(95)	89(105)	300(345)	580(641)	0/15
ABC	2.3(1)	10(6)	95(103)	∞	∞	∞ *5e5*	0/15
GA	**2.1**(2)	84(59)	93(99)	148(170)	621(690)	605(726)	0/15
f17	5.2	215	899	3669	6351	7934	15/15
ACOR	**3.1**(3)	**1.8**(0.4)$^{\star 2}$	**0.95**(0.2)$^{\star 4}$	**2.8**(8)$^{\star 3}$	**7.5**(9)$^{\star 3}$	**11**(14)$^{\star 3}$	15/15
PSO	3.3(4)	169(2)	142(279)	548(681)	514(630)	420(498)	1/15
ABC	6.5(7)	15(15)	64(52)	∞	∞	∞ *5e5*	0/15
GA	5.4(6)	46(12)	36(4)	189(226)	550(591)	∞ *5e5*	0/15
f18	103	378	3968	9280	10905	12469	15/15
ACOR	**1.9**(1)	**2.4**(1)$^{\star 2}$	**5.8**(14)$^{\star 2}$	**36**(38)$^{\star 3}$	**82**(70)	**80**(60)	15/15
PSO	2.2(2)	6.6(5)	113(134)	∞	∞	∞ *5e5*	0/15
ABC	5.0(5)	27(25)	300(315)	∞	∞	∞ *5e5*	0/15
GA	22(16)	59(15)	34(64)	∞	∞	∞ *5e5*	0/15
f19	1	1	242	1.2e5	1.2e5	1.2e5	15/15
ACOR	**28**(28)	3135(4608)	**626**(611)	**17**(38)	**17**(38)	**17**(38)	14/15
PSO	35(30)	3381(2858)	2448(3097)	60(69)	61(70)	61(61)	0/15
ABC	34(46)	**2898**(2014)	3823(4341)	∞	∞	∞ *5e5*	0/15
GA	35(23)	1.2e4(7300)	699(490)	60(69)	∞	∞ *5e5*	0/15
f20	16	851	38111	54470	54861	55313	14/15
ACOR	**6.0**(2)	3.2(4)	3.3(4)	2.3(3)	2.3(3)	**2.3**(3)	15/15
PSO	8.7(5)	3.1(1)	27(33)	19(23)	19(23)	18(23)	5/15
ABC	7.2(4)	**1.5**(1)	**0.55**(0.5)	**0.58**(0.4)	**1.5**(1)	2.6(2)	15/15
GA	47(36)	21(4)	1(0.2)	1.3(0.2)	2.6(0.4)	5.0(5)	11/15
f21	41	1157	1674	1705	1729	1757	14/15
ACOR	3.8(4)	118(196)	299(212)	294(209)	290(206)	285(202)	15/15
PSO	**2.0**(2)	379(434)	262(448)	258(440)	255(291)	**252**(286)	8/15
ABC	3.2(2)	**1.8**(2)	**6.7**(8)	**13**(13)	**84**(108)	265(350)	8/15
GA	4.6(5)	5.5(4)	61(150)	70(148)	139(154)	291(428)	8/15
f22	71	386	938	1008	1040	1068	14/15
ACOR	2.9(3)	143(261)	855(1544)	797(1437)	774(1394)	756(1359)	15/15
PSO	**2.6**(2)	325(647)	469(535)	439(744)	**429**(721)	**422**(702)	8/15
ABC	5.1(5)	**7.6**(11)	**35**(44)	**374**(501)	3311(3605)	6900(7495)	0/15
GA	6.0(6)	18(12)	388(546)	1489(1598)	6830(7723)	∞ *5e5*	0/15
f23	3.0	518	14249	31654	33030	34256	15/15
ACOR	2.6(3)	86(79)	∞	∞	∞	∞ *1e7*	0/15
PSO	2.2(2)	20(17)	**243**(254)*	∞	∞	∞ *5e5*	0/15
ABC	2.2(2)	**19**(16)	∞	∞	∞	∞ *5e5*	0/15
GA	**1.5**(1)	59(61)	∞	∞	∞	∞ *5e5*	0/15
f24	1622	2.2e5	6.4e6	9.6e6	1.3e7	1.3e7	3/15
ACOR	8.2(6)	**155**(169)	∞	∞	∞	∞ *1e7*	0/15
PSO	**5.7**(5)	∞	∞	∞	∞	∞ *5e5*	0/15
ABC	13(18)	∞	∞	∞	∞	∞ *5e5*	0/15
GA	21(9)	∞	∞	∞	∞	∞ *5e5*	0/15

Table 1: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 (given in the respective first row) for different Δf values in dimension 5. The central 80% range divided by two is given in braces. The median number of conducted function evaluations is additionally given in *italics*, if $\mathrm{ERT}(10^{-7}) = \infty$. #succ is the number of trials that reached the final target $f_{\mathrm{opt}} + 10^{-8}$. Best results are printed in bold.

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f1	43	43	43	43	43	43	15/15
ACOR	24(5)	**43**(8)	**60**(8)	**95**(9)$^{\star 2}$	**129**(10)$^{\star 4}$	**164**(8)$^{\star 4}$	15/15
PSO	**22**(8)	3399(39)	3446(31)	3563(37)	3680(41)	3808(48)	14/15
ABC	36(24)	66(41)	94(56)	177(69)	292(17)	374(23)	15/15
GA	876(65)	1905(154)	3205(241)	3.1e4(5e4)	6.7e5(7e5)	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f2	385	386	387	390	391	393	15/15
ACOR	**10**(0.9)	**12**(0.7)$^{\star 3}$	**14**(0.5)$^{\star 4}$	**17**(1)$^{\star 4}$	**21**(0.9)$^{\star 4}$	**25**(0.9)$^{\star 4}$	15/15
PSO	4580(7798)	4572(5190)	4571(7763)	4547(7700)	4542(7666)	4533(7626)	8/15
ABC	12(5)	16(4)	24(6)	40(8)	56(6)	70(6)	15/15
GA	3231(2721)	6772(7870)	7.3e4(9e4)	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f3	5066	7626	7635	7643	7646	7651	15/15
ACOR	1018(1197)	∞	∞	∞	∞	∞ *1e7*	0/15
PSO	∞	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	**1.5**(0.8)$^{\star 4}$	**2.7**(1)$^{\star 4}$	**3.0**(2)$^{\star 4}$	**3.6**(2)$^{\star 4}$	**4.1**(1)$^{\star 4}$	**4.9**(0.8)$^{\star 4}$	15/15
GA	29(5)	3709(4068)	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f4	4722	7628	7666	7700	7758	1.4e5	9/15
ACOR	1.5e4(1e4)	∞	∞	∞	∞	∞ *1e7*	0/15
PSO	5940(6989)	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	**2.0**(1)$^{\star 4}$	**4.2**(2)$^{\star 4}$	**5.6**(4)$^{\star 4}$	**6.3**(4)$^{\star 4}$	**7.3**(4)$^{\star 4}$	**0.45**(0.2)$^{\star 4}$	15/15
GA	65(5)	3751(4198)	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f5	41	41	41	41	41	41	15/15
ACOR	**10**(3)$^{\star 2}$	**12**(4)$^{\star 2}$	**12**(4)$^{\star 3}$	**12**(4)$^{\star 4}$	**12**(4)$^{\star 4}$	**12**(4)$^{\star 4}$	15/15
PSO	4.3e4(5e4)	4.3e4(5e4)	4.3e4(7e4)	4.3e4(7e4)	4.3e4(5e4)	4.3e4(5e4)	8/15
ABC	69(23)	90(36)	93(38)	93(36)	93(36)	93(36)	15/15
GA	2158(156)	4592(305)	7519(403)	1.4e4(574)	2.2e4(817)	2.3e5(2e5)	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f6	1296	2343	3413	5220	6728	8409	15/15
ACOR	**6.5**(2)$^{\star 3}$	**5.3**(1)$^{\star 4}$	**4.8**(0.9)$^{\star 4}$	**4.7**(0.6)$^{\star 4}$	**4.8**(0.6)$^{\star 4}$	**4.9**(0.6)$^{\star 4}$	15/15
PSO	1082(1566)	1009(1291)	705(884)	502(604)	423(485)	577(631)	5/15
ABC	46(54)	453(588)	2587(2687)	∞	∞	∞ *2e6*	0/15
GA	1956(2329)	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f7	1351	4274	9503	16524	16524	16969	15/15
ACOR	**76**(36)	**3.5e4**(3e4)	∞	∞	∞	∞ *1e7*	0/15
PSO	427(745)	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	251(328)	∞	∞	∞	∞	∞ *2e6*	0/15
GA	77(15)	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f8	2039	3871	4040	4219	4371	4484	15/15
ACOR	24(8)	66(114)	72(101)	**89**(86)	**107**(84)$^{\star 3}$	**125**(83)$^{\star 4}$	15/15
PSO	90(64)	307(335)	350(329)	466(348)	894(743)	3277(3290)	1/15
ABC	**3.9**(2)$^{\star 4}$	**5.9**(2)$^{\star 4}$	**10**(3)$^{\star 4}$	353(367)	∞	∞ *2e6*	0/15
GA	∞	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f9	1716	3102	3277	3455	3594	3727	15/15
ACOR	**45**(7)$^{\star 3}$	**78**(8)$^{\star 4}$	**119**(11)$^{\star 4}$	**286**(31)$^{\star 4}$	**473**(52)$^{\star 4}$	**651**(74)	15/15
PSO	671(437)	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	699(596)	∞	∞	∞	∞	∞ *2e6*	0/15
GA	∞	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f10	7413	8661	10735	14920	17073	17476	15/15
ACOR	**2.0e4**(2e4)	∞	∞	∞	∞	∞ *1e7*	0/15
PSO	∞	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	∞	∞	∞	∞	∞	∞ *2e6*	0/15
GA	∞	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f11	1002	2228	6278	9762	12285	14831	15/15
ACOR	1002(53)	1328(2262)	679(809)	995(1040)	1037(1426)	**1462**(1688)	2/15
PSO	**143**(47)$^{\star 3}$	**186**(29)	**109**(12)	**132**(12)	**148**(16)	2019(2158)	0/15
ABC	∞	∞	∞	∞	∞	∞ *2e6*	0/15
GA	2.9e4(3e4)	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f12	1042	1938	2740	4140	12407	13827	15/15
ACOR	3665(4801)	1.0e4(1e4)	2.4e4(3e4)	∞	∞	∞ *1e7*	0/15
PSO	1704(1934)	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	**26**(5)	**67**(50)	**429**(429)*	**7243**(7971)$^{\star 2}$	∞	∞ *2e6*	0/15
GA	∞	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f13	652	2021	2751	18749	24455	30201	15/15
ACOR	1.8e4(2e4)	3.2e4(4e4)	5.1e4(6e4)	∞	∞	∞ *1e7*	0/15
PSO	6156(7673)	6441(7421)	1.0e4(1e4)	**1495**(1600)	∞	∞ *2e6*	0/15
ABC	**25**(11)	**73**(81)*	**627**(735)$^{\star 2}$	1555(1787)	∞	∞ *2e6*	0/15
GA	5116(6184)	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f14	75	239	304	932	1648	15661	15/15
ACOR	15(4)	**9.5**(2)	**11**(2)$^{\star 3}$	**30**(8)$^{\star 3}$	∞	∞ *1e7*	0/15
PSO	**6.7**(3)*	12(3)	20(3)	54(14)	∞	∞ *2e6*	0/15
ABC	18(16)	18(11)	28(8)	3378(3218)	∞	∞ *2e6*	0/15
GA	277(74)	319(48)	477(66)	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f15	30378	1.5e5	3.1e5	3.2e5	4.5e5	4.6e5	15/15
ACOR	**919**(1161)	∞	∞	∞	∞	∞ *1e7*	0/15
PSO	∞	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	∞	∞	∞	∞	∞	∞ *2e6*	0/15
GA	∞	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f16	1384	27265	77015	1.9e5	2.0e5	2.2e5	15/15
ACOR	1.0e5(1e5)	∞	∞$^{\star 4}$	∞$^{\star 4}$	∞$^{\star 4}$	∞ *1e7*$^{\star 4}$	0/15
PSO	111(11)	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	**13**(14)	∞	∞	∞	∞	∞ *2e6*	0/15
GA	138(40)	**1037**(1138)	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f17	63	1030	4005	30677	56288	80472	15/15
ACOR	12(8)	**4.5**(1)$^{\star 4}$	**269**(447)$^{\star 3}$	**4564**(5053)	∞	∞ *1e7*	0/15
PSO	**3.2**(2)$^{\star 3}$	2502(2914)	∞	∞	∞	∞ *2e6*	0/15
ABC	34(27)	∞	∞	∞	∞	∞ *2e6*	0/15
GA	57(54)	92(8)	7070(7745)	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f18	621	3972	19561	67569	1.3e5	1.5e5	15/15
ACOR	**4.7**(2)	**3.5**(1)$^{\star 4}$	**493**(607)	∞	∞	∞ *1e7*	0/15
PSO	236(9)	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	4.6e4(5e4)	∞	∞	∞	∞	∞ *2e6*	0/15
GA	76(15)	311(506)	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f19	1	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
ACOR	686(252)	∞	∞	∞	∞	∞ *1e7*	0/15
PSO	**382**(240)$^{\star 2}$	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	2292(1142)	∞	∞	∞	∞	∞ *2e6*	0/15
GA	1.4e4(3105)	**6.5e5**(1e5)$^{\star 4}$	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f20	82	46150	3.1e6	5.5e6	5.6e6	5.6e6	14/15
ACOR	**16**(4)	1.6(1)	**24**(24)	**27**(28)	**26**(30)	**26**(28)	1/15
PSO	17(6)	50(65)	∞	∞	∞	∞ *2e6*	0/15
ABC	16(5)	**0.12**(0.1)$^{\star 3}_{\downarrow 4}$	∞	∞	∞	∞ *2e6*	0/15
GA	503(56)	2.8(0.4)	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f21	561	6541	14103	14643	15567	17589	15/15
ACOR	2743(8906)	9938(1e4)	9927(1e4)	9561(1e4)	8994(1e4)	7960(9097)	1/15
PSO	1784(3563)	4281(5198)	1986(2340)	1913(2117)	1799(1863)	1593(1762)	1/15
ABC	**5.0**(3)	**22**(50)*	**25**(30)*	**27**(30)$^{\star 2}$	**35**(37)$^{\star 2}$	**85**(96)$^{\star 2}$	8/15
GA	90(19)	625(766)	398(496)	904(1093)	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f22	467	5580	23491	24948	26847	1.3e5	12/15
ACOR	1.4e4(2e4)	3585(5376)	∞	∞	∞	∞ *1e7*	0/15
PSO	**5.0**(5)*	411(539)	∞	∞	∞	∞ *2e6*	0/15
ABC	10(6)	**47**(67)	**77**(86)$^{\star 2}$	∞	∞	∞ *2e6*	0/15
GA	110(53)	1452(1615)	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f23	3.2	1614	67457	4.9e5	8.1e5	8.4e5	15/15
ACOR	1.8(2)	∞	∞$^{\star 3}$	∞$^{\star 3}$	∞$^{\star 3}$	∞ *1e7*$^{\star 3}$	0/15
PSO	2.2(2)	1554(1724)	∞	∞	∞	∞ *2e6*	0/15
ABC	**1.1**(0.5)	**78**(66)$^{\star 3}$	∞	∞	∞	∞ *2e6*	0/15
GA	1.7(2)	4570(4340)	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
f24	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
ACOR	∞	∞	∞	∞	∞	∞ *1e7*	0/15
PSO	∞	∞	∞	∞	∞	∞ *2e6*	0/15
ABC	∞	∞	∞	∞	∞	∞ *2e6*	0/15
GA	∞	∞	∞	∞	∞	∞ *2e6*	0/15

Table 2: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 (given in the respective first row) for different Δf values in dimension 20. The central 80% range divided by two is given in braces. The median number of conducted function evaluations is additionally given in *italics*, if $\mathrm{ERT}(10^{-7}) = \infty$. #succ is the number of trials that reached the final target $f_{\mathrm{opt}} + 10^{-8}$. Best results are printed in bold.

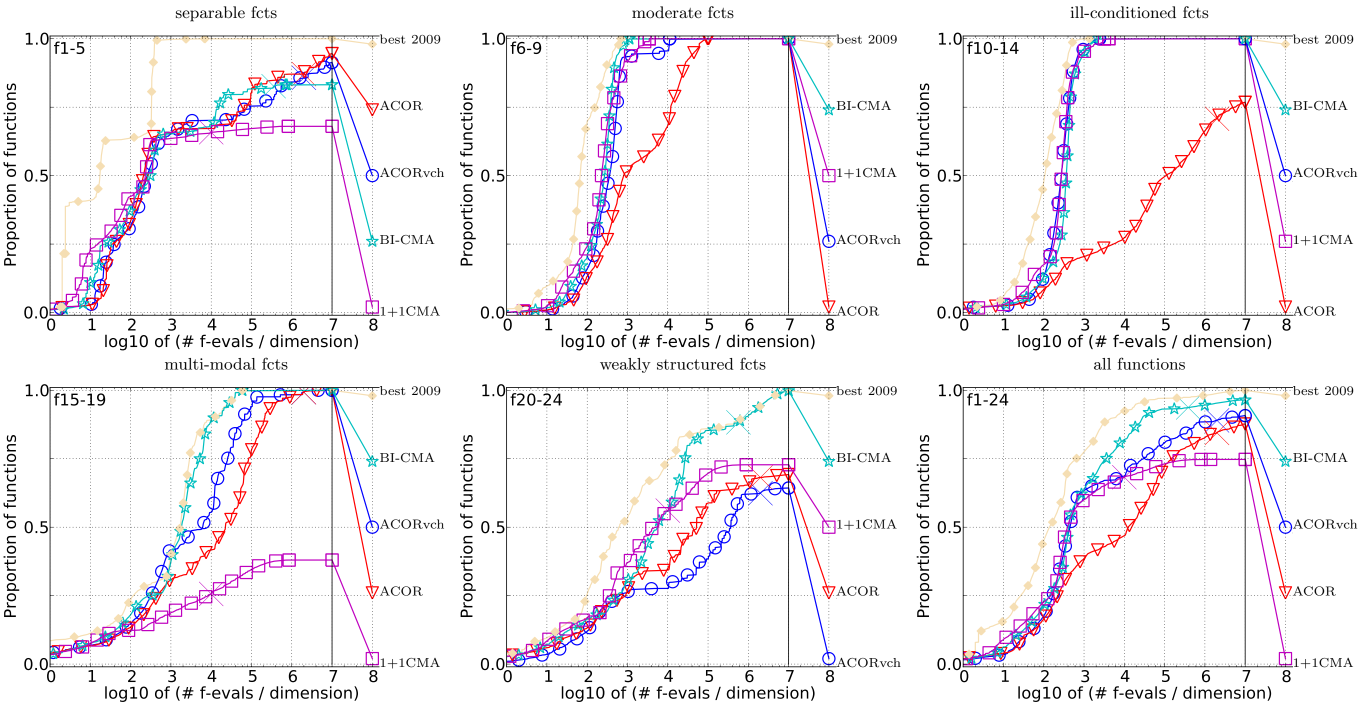

Figure 4: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/D) for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 5-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single target.

[2] M. Dorigo. *Optimization, Learning and Natural Algorithms* (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.

[3] M. El-Abd. Black-box optimization benchmarking for noiseless function testbed using artificial bee colony algorithm. In *Proceedings of Conference on Genetic and Evolutionary Computation, GECCO'10*, pages 1719–1724, New York, NY, USA, 2010. ACM.

[4] M. El-Abd and M. S. Kamel. Black-box optimization benchmarking for noiseless function testbed using particle swarm optimization. In *Proceedings of Conference on Genetic and Evolutionary Computation, GECCO'09*, pages 2269–2274, New York, NY, USA, 2009. ACM.

[5] 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.

[6] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'09*, pages 2389–2396, New York, NY, USA, 2009. ACM.

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

[8] 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.

[9] G. Leguizamón and C. Coello. An alternative acor algorithm for continuous optimization problems. In M. Dorigo et al., editors, *Proceedings of the Seventh International Conference on Swarm Intelligence, ANTS 2010*, volume 6234 of *LNCS*, pages 48–59. Springer, Berlin, Germany, 2010.

[10] T. Liao, D. Molina, M. Montes de Oca, and T. Stützle. A Note on the Effects of Enforcing Bound Constraints on Algorithm Comparisons using the IEEE CEC'05 Benchmark Function Suite. Technical Report TR/IRIDIA/2011-010, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.

[11] T. Liao, M. A. Montes de Oca, D. Aydin, T. Stützle, and M. Dorigo. An incremental ant colony algorithm with local search for continuous optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'11*, pages 125–132, New York, NY, USA, 2011. ACM.

[12] M. Nicolau. Application of a simple binary genetic algorithm to a noiseless testbed benchmark. In *Proceedings of Conference on Genetic and Evolutionary Computation, GECCO'09*, pages 2473–2478, New York, NY, USA, 2009. ACM.

[13] K. Socha and M. Dorigo. Ant colony optimization for continuous domains. *European Journal of Operational Research*, 185(3):1155–1173, 2008.