Optimization Benchmarking for A2 algorithm

*

Guillaumme Lorre Abdallah Benzine Mahmoud Loukkal

Hafed Rhouma Mohamed Abdel Elkhalek

ABSTRACT

The aim of COCO Platform is to benchmark continuous algorithms. Here we deal with single objective functions, and we benchmark two implementations of A2 aglorithm (coded with python and C languages) dealing with the evolution strategies issues with 2 other performing algorithms BIPOP-C and BFGS.

Keywords

Benchmarking, Black-box optimization, A2

1. INTRODUCTION

Inspired by biology and Darwin´ s , evolutions strategies ES are stochastic optimization algorithms designed for continuous search space. Unlike the gradient based algorithms which are local search algorithms, ES are global optimization algorithms and perform very well on difficult problems such as badly scaled, non-continuously differentiable or even not completely defined functions (Blackbox). [2, 12] ES are heuristic population-based search algorithm that incorporate random variation and selection. In each iteration called generation, ES algorithm generates offsprings from μ parents. The offsprings are generated by adding a mutation vector to the parents. The mutation vectors are Gaussian distributed with mean equal to zero and standard deviation σ called step size. Then environmental selection reduces the population to its original size. [2, 12] The important features of ES are:

- Unbiasedness: the mutation is based on normally distributed vectors so the information injected at each generation is unbiased.
- Self-adaptation:strategy parameter control, step size adaptation.
- Environmental selection: survival of the fittest, unlike the genetic algorithms where the individuals are randomly selected, the selection process in ES is deterministic.

Various step size adaptation concepts have been imagined since the creation of ES algorithms and perform well but when the scaling of the parameters to be optimized is not known, the idea of individual step sizes has to be implemented. Schwefel and Rechenberg have introduced the idea of mutative step size control. This concept is based on the idea that both objective parameters (solution, individual) and strategy parameters (here step size) undergo mutation and selection. One shortcoming of the adaptation of individual step sizes is that it is impossible for small population. This is due to the fact that the size of the parameter variation is not taken into account. For example, the object parameter can undergo a large variation even if the step size variation is small because the randomly generated vector (the one added to mutate the offspring) is large. Another reason is that the step size variation between the offsprings of a generation is the same as the one from between generations. This makes the individual step size irrelevant [10]. This problem has been addressed by introducing the derandomized mutative step-size control. The algorithm studied A2 is a good example of the derandomization.

2. ALGORITHM PRESENTATION

In this part, we use the same notations than in [8] and we make reference to some equations of this paper.
A2 is a (μ , λ) evolutionary strategy. The algorithm takes place in three different steps : the mutation of object variables, the adaptation of strategy parameters and the selection[8].

2.1 Mutation

The mutation is done from a parent randomly chosen (E_k) to produce a new individual (N_k). The equation of the mutation is :

$$x_i^{N^k} = x_i^{E_\xi} + \delta_i^{E_k}.z_i^k + \delta_r^{E_k}.z_r^k.r_i^{E_\xi} \tag{1}$$

This mutation can be decomposed in 2 parts. The individual mutation (independant for each vector's component) and the global mutation in one direction (r).
We make this mutation on μ parents for each generation.

*

GECCO'13, July 6-10, 2013, Amsterdam, The Netherlands.

ACM ISBN TBA.

DOI: 10.1145/1235

This mutations needs to be adapted, which means that the mutations which were good in the past have a higher probability to be chosen.

2.2 Mutation adaptation

There are two types of adaptation: individual step size adaptation and direction adaptation.
Individual step size adaptation enables different variances of the mutation on each axis and consequently to have good results on ill-conditioned functions. However, the mutation is dependant of the coordinate system[8].

$$\delta_i^N = \delta_i^E.exp(\beta(\|s^N\| - \chi_n)).exp(\beta_{ind}(|s_i^N| - \chi_1)) \quad (2)$$

The direction adaptation is done by accumulation which makes it possible to keep the information of previous mutations.

$$r' = (1 - c_r).\delta_r^{E_\xi}.r^{E_\xi} + c_r.(x^{N_k} - x^{E_\xi}) \quad (3)$$

Derandomization.

These two adaptations are derandomized which reduces the stochastic noise of the procedure.
1^{st} **derandomization**

It is done by adding β and β_{ind} in the formula of δ. If they are inferior to one, it reduces the evolution of the parameters without changing the strength of the mutation[8, 7].
2^{nd} **derandomization**

In the formulas, it is represented by : $\|s^N\| - \chi_n$ and $|s_i^N| - \chi_1$
If s is greater than his expectation and that the mutation is efficient (it means that the son beget is present in the next generation) then, δ needs to be increased[8, 7].
S represents the accumulation of the mutations. It is better to use accumulation than only the last z_i because our adaptation will be based on all the previous mutations. Accumulation indicates if a sequence of mutations is efficient (the mutations must go in the same direction).[8, 7]

2.3 Selection

The selection is the step in which the algorithm selects the μ best individuals from the offspring to produce the new generation.

3. ALGORITHM IMPLEMENTATION

The algorithm was implemented in Python. First, a class Individu was created which contains as attributes the object and strategy parameters :

- X : numpy vector of the object variables
- Delta : numpy vector of individual step sizes
- DeltaR : step-size for direction
- R : numpy direction vector
- S : accumulation of the realized vectors z
- Sr : weighted sum of DeltaR

. It also contains the method mutation which from a parent creates a child.
The second class Population is constituted of an array of individuals. It contains the following attributes :

- F : the function to be optimized
- N : problem dimensionality
- Lan : number of parents
- Mu : number of children
- Individus : python list containing the population

The method *nextgen* creates the next generation.

4. RESULTS

Results from experiments according to [9] and [3] on the benchmark functions given in [1, 6] are presented in Figures 1, 2 and 3 and in Tables 1 and 2. The experiments were performed with COCO [5], version 1.0.1, the plots were produced with version 1.0.4.

The **average runtime (aRT)**, used in the figures and tables, 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 [4, 11]. **Statistical significance** is tested with the rank-sum test for a given target Δf_t 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.

Figure 1 :

Globally, we observe that the arT in number of f-evaluations is better for the BFGS algortithm on all the functions. Alsoit has the best statistical result compared to all other algorithms. Zoubab and A2 algorithm are competitive regarding the arT.Zoubab does better for Sphere and Ellipsoid Separable algorithms but for the rest they similar results.

Figure 2 :

Our algorithm perfoms rather correctly compared to the others. The BIPOP-C alogorithm is definitely the best one. So if we compare to Zoubab, we have better results, and compared to BFGS, it has better results for weakly structured multi-modalfunctions but for the rest we are either better or equal.

Figure 3 :

For the 20-D the trends are differents. A2 and carepediem Algorithms are globally performing equally. Regarding BFGS it does better for weakly structured multi-modal functions (weakly) and we have better resluts for multi-modal functions.

5. CONCLUSION

In this project, we implemented A2 evolution strategy using python programming language, and compared its performance with the other algorithms (CMA-ES, BFGSâĂę) and the implementation of the same algorithm with C language by team Zoubab. The results figures illustrate the fact that the average running time of our algorithm is lesser than the other teamâĂŹs algorithm (Zoubab) in most of the cases. Even if it is quite performant, our algorithm is still not as effective the CMA-ES, which was predictable. We will analyse our results thoroughly and present our final conclusion during the oral presentation of our work

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, D. V. Arnold, and A. Auger. Evolution strategies. In *Springer Handbook of Computational Intelligence*, pages 871–898. Springer, 2015.

[3] N. Hansen, A. Auger, D. Brockhoff, D. Tušar, and T. Tušar. COCO: Performance assessment. *ArXiv e-prints*, arXiv:1605.03560, 2016.

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

[5] N. Hansen, A. Auger, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *ArXiv e-prints*, arXiv:1603.08785, 2016.

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

[8] N. Hansen, A. Ostermeier, and A. Gawelczyk. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In *ICGA*, pages 57–64, 1995.

[9] N. Hansen, T. Tušar, O. Mersmann, A. Auger, and D. Brockhoff. COCO: The experimental procedure. *ArXiv e-prints*, arXiv:1603.08776, 2016.

[10] A. Ostermeier, A. Gawelczyk, and N. Hansen. A derandomized approach to self-adaptation of evolution strategies. *Evolutionary Computation*, 2(4):369–380, 1994.

[11] 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.

[12] R. Salomon. Evolutionary algorithms and gradient search: similarities and differences. *IEEE Transactions on Evolutionary Computation*, 2(2):45–55, 1998.

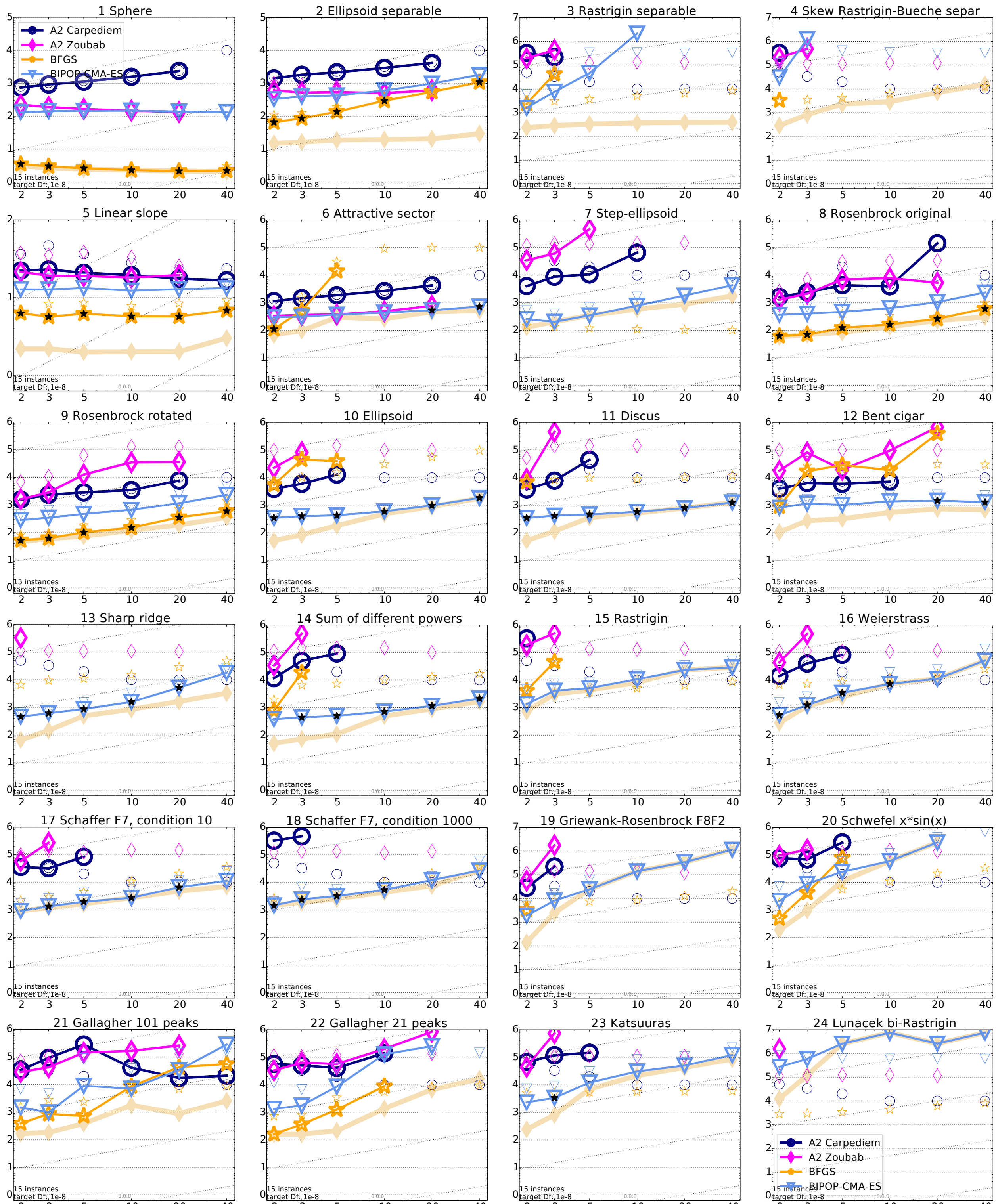

Figure 1: Average running time (aRT in number of f-evaluations as $\log_{10}$ value), divided by dimension for target function value 10^{-8} versus dimension. Slanted grid lines indicate quadratic scaling with the dimension. Different symbols correspond to different algorithms given in the legend of f_1 and f_{24}. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Black stars indicate a statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: ○: A2 Carpediem, ◇: A2 Zoubab, ⋆: BFGS, ▽: BIPOP-CMA-ES

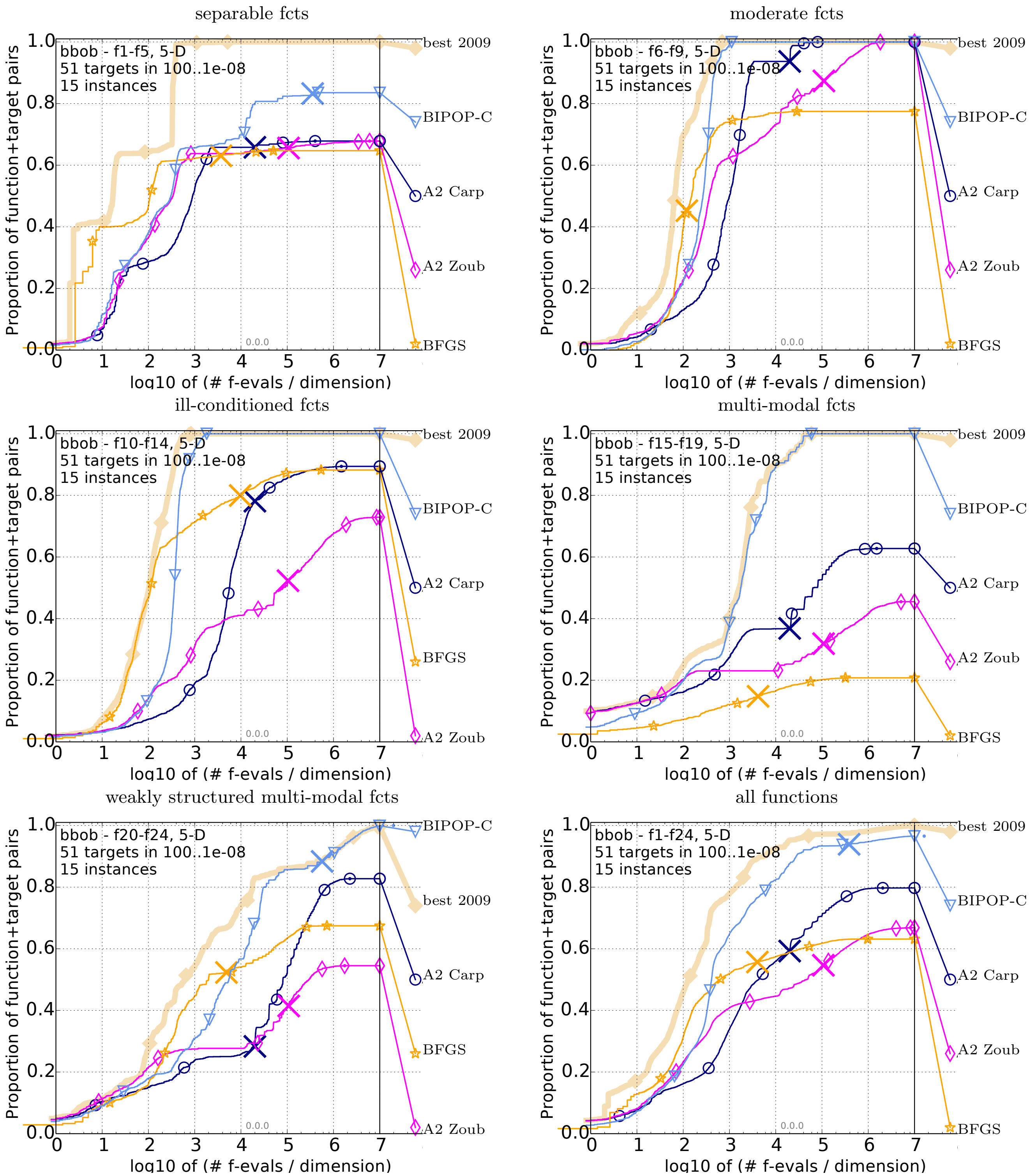

Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 51 targets with target precision in $10^{[-8..2]}$ for all functions and subgroups in 5-D. The "best 2009" line corresponds to the best aRT observed during BBOB 2009 for each selected target.

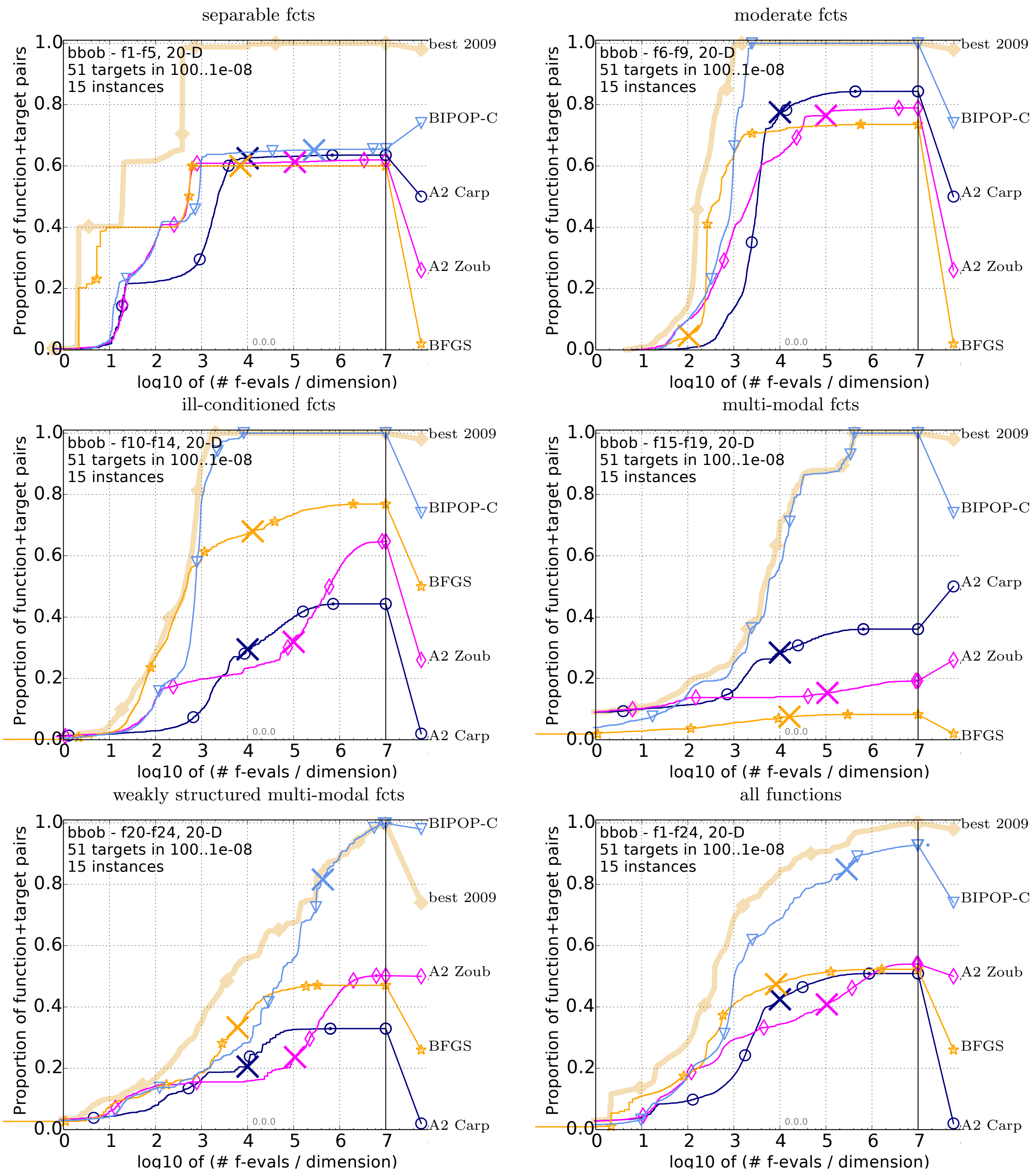

Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 51 targets with target precision in $10^{[-8..2]}$ for all functions and subgroups in 20-D. The "best 2009" line corresponds to the best aRT observed during BBOB 2009 for each selected target.

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f1	11	12	12	12	12	12	12	15/15
A2 Carp	2.1(1)	11(6)	50(7)	107(26)	167(26)	283(24)	391(32)	15/15
A2 Zoub	2.2(2)	7.9(3)	15(4)	22(5)	30(5)	45(4)	60(9)	15/15
BFGS	**1.2**(0)	**1.1**(0)^{*4}	**1.1**(0)^{*4}	**1.1**(0)^{*4}	**1.1**(0)^{*4}	**1.1**(0)^{*4}	**1.1**(0)^{*4}	15/15
BIPOP-C	3.2(2)	9.0(3)	15(3)	21(2)	27(4)	40(3)	53(5)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f2	83	87	88	89	90	92	94	15/15
A2 Carp	25(3)	32(3)	42(6)	53(4)	65(3)	86(4)	103(4)	15/15
A2 Zoub	15(5)	19(7)	21(4)	22(8)	23(5)	26(7)	28(9)	15/15
BFGS	**3.8**(3)^{*4}	**5.6**(2)^{*4}	**6.2**(2)^{*4}	**6.5**(1)^{*4}	**6.6**(1)^{*4}	**6.9**(2)^{*4}	**7.1**(1)^{*4}	15/15
BIPOP-C	13(4)	16(3)	18(1)	19(2)	20(2)	21(3)	22(2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f3	716	1622	1637	1642	1646	1650	1654	15/15
A2 Carp	25(1)	404(385)	∞	∞	∞	∞	∞ *1e5*	0/15
A2 Zoub	203(632)	4781(6390)	∞	∞	∞	∞	∞ *5e5*	0/15
BFGS	107(71)	∞	∞	∞	∞	∞	∞ *2e4*	0/15
BIPOP-C	**1.4**(0.9)	**16**(11)	**139**(65)	**139**(521)	**139**(563)	**139**(110)	**140**(305)	14/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f4	809	1633	1688	1758	1817	1886	1903	15/15
A2 Carp	34(0.8)	∞	∞	∞	∞	∞	∞ *1e5*	0/15
A2 Zoub	170(206)	∞	∞	∞	∞	∞	∞ *5e5*	0/15
BFGS	169(147)	∞	∞	∞	∞	∞	∞ *2e4*	0/15
BIPOP-C	**2.7**(1)	∞	∞	∞	∞	∞	∞ *2e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f5	10	10	10	10	10	10	10	15/15
A2 Carp	7.7(4)	10(4)	10(4)	10(5)	10(6)	10(5)	10(4)	15/15
A2 Zoub	7.3(5)	9.5(3)	9.5(4)	9.5(4)	9.5(5)	9.5(4)	9.5(5)	15/15
BFGS	**1.9**(0.5)^{*3}	**3.0**(0.9)^{*3}	**3.1**(0.3)^{*3}	**3.1**(0.8)^{*3}	**3.1**(0.5)^{*3}	**3.1**(0.5)^{*3}	**3.1**(1)^{*3}	15/15
BIPOP-C	4.5(1)	6.5(2)	6.6(2)	6.6(2)	6.6(2)	6.6(2)	6.6(3)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f6	114	214	281	404	580	1038	1332	15/15
A2 Carp	2.0(0.6)	5.7(1)	7.4(2)	7.9(0.8)	7.3(0.8)	6.0(0.3)	6.3(0.3)	15/15
A2 Zoub	**1.7**(0.8)	**1.6**(0.3)	**1.8**(0.6)	**1.8**(0.4)	**1.6**(0.3)	**1.2**(0.3)	**1.3**(0.2)	15/15
BFGS	3.0(2)	3.3(1)	3.4(2)	3.0(1.0)	2.5(1)	2.0(0.8)	7.8(7)	15/15
BIPOP-C	2.3(1.0)	2.1(0.6)	2.2(0.7)	1.9(0.4)	1.7(0.2)	1.3(0.2)	1.3(0.1)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f7	24	324	1171	1451	1572	1572	1597	15/15
A2 Carp	**2.9**(3)	2.0(2)	7.6(0.3)	19(35)	25(32)	25(32)	25(47)	10/15
A2 Zoub	3.6(9)	577(1002)	658(1149)	542(874)	1112(1394)	1112(1652)	1471(1975)	3/15
BFGS	∞	∞	∞	∞	∞	∞	∞ *600*	0/15
BIPOP-C	5.0(4)	**1.5**(1)	**1**(1)	**1**(0.2)	**1**(0.7)	**1**(0.9)	**1**(0.9)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f8	73	273	336	372	391	410	422	15/15
A2 Carp	7.8(2)	35(2)	36(2)	37(68)	37(3)	42(123)	48(6)	14/15
A2 Zoub	4.0(2)	54(86)	50(12)	52(122)	55(69)	64(37)	76(48)	15/15
BFGS	**2.1**(1)	**1.8**(3)^{*2}	**1.6**(2)^{*2}	**1.5**(1)^{*3}	**1.5**(0.4)^{*3}	**1.5**(0.4)^{*3}	**1.5**(0.2)^{*3}	15/15
BIPOP-C	3.2(2)	3.7(3)	4.5(0.6)	4.7(1)	4.8(2)	5.1(4)	5.4(3)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f9	35	127	214	263	300	335	369	15/15
A2 Carp	18(9)	18(8)	22(2)	25(6)	26(6)	31(7)	34(6)	15/15
A2 Zoub	4.5(2)	243(646)	150(75)	133(115)	128(284)	140(93)	156(416)	15/15
BFGS	**3.6**(3)	**3.0**(0.7)^{*2}	**2.0**(1)^{*3}	**1.8**(0.7)^{*3}	**1.6**(0.5)^{*3}	**1.5**(0.8)^{*4}	**1.4**(0.6)^{*4}	15/15
BIPOP-C	5.8(1)	8.7(4)	7.2(2)	6.7(5)	6.4(4)	6.3(4)	6.2(5)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f10	349	500	574	607	626	829	880	15/15
A2 Carp	42(18)	42(18)	43(14)	49(16)	56(15)	56(14)	68(20)	15/15
A2 Zoub	241(907)	679(850)	1446(1199)	6141(6233)	5960(1e4)	4513(7623)	9469(7777)	0/15
BFGS	**1**(0.3)^{*4}	**1**(0.3)^{*4}	**1**(0.2)^{*4}	**1**(0.2)^{*4}	**1**(0.4)^{*4}	**1.1**(0.2)^{*2}	23(39)	5/15
BIPOP-C	3.5(0.6)	2.9(0.5)	2.7(0.4)	2.7(0.3)	2.8(0.3)	2.3(0.2)	**2.4**(0.2)^{*2}	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f11	143	202	763	977	1177	1467	1673	15/15
A2 Carp	138(61)	143(63)	54(31)	62(25)	60(69)	89(55)	112(142)	6/15
A2 Zoub	3236(3517)	1.8e4(2e4)	∞	∞	∞	∞	∞ *5e5*	0/15
BFGS	**1**(0.2)^{*4}	**1**(0.1)^{*4}	**1.1**(0.8)*	1.9(2)	8.2(4)	199(272)	∞ *4e4*	0/15
BIPOP-C	8.4(3)	7.2(2)	2.2(0.3)	**1.8**(0.2)	**1.6**(0.2)*	**1.4**(0.1)^{*4}	**1.3**(0.1)^{*4}	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f12	108	268	371	413	461	1303	1494	15/15
A2 Carp	73(37)	39(25)	35(15)	39(9)	42(11)	18(3)	19(2)	15/15
A2 Zoub	165(593)	210(5)	154(513)	139(311)	166(409)	73(98)	64(85)	14/15
BFGS	**1.1**(1)^{*4}	**1**(0.5)^{*4}	**1**(0.5)^{*3}	**1**(0.4)^{*4}	**1**(0.7)^{*4}	**2.0**(3)	49(92)	5/15
BIPOP-C	11(12)	7.4(8)	7.4(6)	7.5(4)	7.7(4)	3.3(2)	**3.3**(2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f13	132	195	250	319	1310	1752	2255	15/15
A2 Carp	15(2)	151(134)	5628(7006)	4408(7749)	∞	∞	∞ *1e5*	0/15
A2 Zoub	70(122)	613(159)	1748(3336)	5354(5269)	2842(1461)	∞	∞ *5e5*	0/15
BFGS	**1**(0.3)^{*4}	**1**(0.1)^{*4}	**1**(0.0)^{*4}	**1**(0.0)^{*4}	4.8(11)	136(93)	∞ *5e4*	0/15
BIPOP-C	3.9(3)	5.4(3)	5.9(2)	5.4(0.9)	**1.6**(0.3)	**1.5**(0.3)^{*4}	**1.7**(0.2)^{*4}	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f14	10	41	58	90	139	251	476	15/15
A2 Carp	1.1(1)	2.4(1)	12(5)	18(2)	23(5)	69(7)	297(386)	3/15
A2 Zoub	**0.95**(0.5)	2.1(1)	3.2(0.9)	3.7(0.5)	5.0(1)	586(1361)	∞ *6e5*	0/15
BFGS	2.2(1)	**1.7**(1)	**1.8**(1)^{*2}	**1.5**(0.7)^{*4}	**1.3**(0.4)^{*4}	**1**(0.2)^{*4}	350(196)	0/15
BIPOP-C	1.1(0.9)	2.8(1)	3.7(0.7)	4.0(1)	4.6(1)	5.4(1)	**4.5**(0.3)^{*4}	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f15	511	9310	19369	19743	20073	20769	21359	14/15
A2 Carp	36(99)	∞	∞	∞	∞	∞	∞ *1e5*	0/15
A2 Zoub	191(282)	860(782)	∞	∞	∞	∞	∞ *5e5*	0/15
BFGS	87(137)	∞	∞	∞	∞	∞	∞ *2e4*	0/15
BIPOP-C	**1.6**(0.7)	**1.5**(1)^{*4}	**1.2**(0.7)	**1.2**(0.6)	**1.2**(0.5)	**1.2**(0.5)	**1.2**(0.5)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f16	120	612	2662	10163	10449	11644	12095	15/15
A2 Carp	3.8(5)	148(287)	105(188)	40(30)	39(31)	35(47)	34(27)	3/15
A2 Zoub	**1.6**(2)	237(357)	658(587)	374(743)	364(587)	∞	∞ *5e5*	0/15
BFGS	153(102)	960(1066)	∞	∞	∞	∞	∞ *4e4*	0/15
BIPOP-C	3.0(4)	**3.6**(3)	**2.6**(1)*	**1.1**(0.7)^{*2}	**1.3**(2)^{*2}	**1.4**(0.6)^{*2}	**1.4**(2)^{*2}	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f17	5.2	215	899	2861	3669	6351	7934	15/15
A2 Carp	**1.6**(2)	1.7(0.9)	2.6(0.3)	4.1(0.2)	6.0(7)	33(20)	52(107)	3/15
A2 Zoub	1.9(2)	58(214)	73(33)	90(148)	347(591)	∞	∞ *5e5*	0/15
BFGS	120(203)	645(679)	∞	∞	∞	∞	∞ *2e4*	0/15
BIPOP-C	3.4(3)	**1**(0.2)	**1**(2)	**1**(1)	**1**(0.6)^{*2}	**1**(0.6)^{*3}	**1.2**(0.4)^{*3}	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f18	103	378	3968	8451	9280	10905	12469	15/15
A2 Carp	1.1(1)	4.0(1)	4.2(0.1)	14(15)	44(40)	∞	∞ *1e5*	0/15
A2 Zoub	79(0.5)	77(193)	78(123)	298(208)	∞	∞	∞ *5e5*	0/15
BFGS	57(96)	∞	∞	∞	∞	∞	∞ *2e4*	0/15
BIPOP-C	**1**(0.8)	**3.4**(3)	**1**(1)	**1**(0.4)	**1**(0.3)^{*2}	**1.2**(0.6)^{*4}	**1.3**(0.7)^{*4}	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f19	1	1	242	1.0e5	1.2e5	1.2e5	1.2e5	15/15
A2 Carp	**1**(0)	**1**(0)	5960(8774)	∞	∞	∞	∞ *1e5*	0/15
A2 Zoub	**1**(0)	**1**(0)	∞	∞	∞	∞	∞ *5e5*	0/15
BFGS	1655(1240)	2.2e4(4e4)	1780(2389)	∞	∞	∞	∞ *3e4*	0/15
BIPOP-C	20(18)	2801(1434)	**161**(161)	**1**(0.9)	**1**(0.9)	**1**(0.7)	**1**(0.7)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f20	16	851	38111	51362	54470	54861	55313	14/15
A2 Carp	2.4(2)	45(30)	37(37)	27(33)	26(60)	26(26)	25(27)	1/15
A2 Zoub	**1.7**(1)	835(790)	∞	∞	∞	∞	∞ *5e5*	0/15
BFGS	1.8(0.9)	**2.5**(2)	10(10)	7.6(10)	7.2(3)	7.1(5)	7.1(8)	1/15
BIPOP-C	3.3(2)	8.2(9)	**2.8**(3)	**2.2**(1)	**2.1**(0.9)	**2.2**(1)	**2.2**(1)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f21	41	1157	1674	1692	1705	1729	1757	14/15
A2 Carp	**2.0**(2)	562(821)	836(926)	828(931)	822(836)	811(795)	798(512)	1/15
A2 Zoub	93(0.7)	217(192)	430(445)	425(544)	422(495)	416(663)	410(534)	8/15
BFGS	3.8(4)	**1.4**(2)	**1.9**(3)	**1.9**(3)	**1.9**(2)	**1.9**(3)	**2.0**(2)	15/15
BIPOP-C	2.3(2)	14(6)	24(74)	25(119)	25(21)	25(36)	25(20)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f22	71	386	938	980	1008	1040	1068	14/15
A2 Carp	218(705)	389(259)	214(240)	206(230)	200(174)	195(120)	192(234)	5/15
A2 Zoub	170(228)	425(545)	304(431)	292(95)	284(371)	275(376)	269(734)	11/15
BFGS	**3.1**(3)	**2.9**(2)	**2.1**(2)	**2.1**(2)	**2.0**(2)	**2.0**(2)	**2.6**(4)	14/15
BIPOP-C	6.9(15)	20(57)	45(105)	43(69)	42(47)	41(84)	40(93)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f23	3.0	518	14249	27890	31654	33030	34256	15/15
A2 Carp	2.3(2)	91(117)	19(15)	12(12)	15(18)	22(15)	21(45)	2/15
A2 Zoub	1.7(2)	95(60)	165(121)	∞	∞	∞	∞ *5e5*	0/15
BFGS	11(4)	31(35)	∞	∞	∞	∞	∞ *2e4*	0/15
BIPOP-C	**1.7**(1)	**13**(9)	**3.7**(5)	**2.1**(2)	**1.8**(1)	**1.8**(1)	**1.8**(2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f24	1622	2.2e5	6.4e6	9.6e6	9.6e6	1.3e7	1.3e7	3/15
A2 Carp	63(79)	∞	∞	∞	∞	∞	∞ *1e5*	0/15
A2 Zoub	50(108)	∞	∞	∞	∞	∞	∞ *5e5*	0/15
BFGS	69(76)	∞	∞	∞	∞	∞	∞ *2e4*	0/15
BIPOP-C	**2.1**(2)	**1.6**(1)	**1**(1)	**1**(0.8)	**1**(1)	**1**(2)	**1**(2)	3/15

Table 1: Average running time (aRT in number of function evaluations) divided by the respective best aRT measured during BBOB-2009 in dimension 5. The aRT and in braces, as dispersion measure, the half difference between 10 and 90%-tile of bootstrapped run lengths appear for each algorithm and target, the corresponding best aRT in the first row. The different target Δf-values are shown in the top row. #succ is the number of trials that reached the (final) target $f_{\text{opt}} + 10^{-8}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction of 110. A ↓ indicates the same tested against the best algorithm of BBOB-2009. Best results are printed in bold.

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f1	43	43	43	43	43	43	43	15/15
A2 Carp	39(12)	159(16)	273(16)	393(22)	507(23)	746(19)	979(23)	15/15
A2 Zoub	5.6(0.8)	12(1)	18(2)	25(2)	31(3)	43(3)	56(2)	15/15
BFGS	**1**(0)$^{\star 4}$	**1**(0)$^{\star 4}$	**1**(0)$^{\star 4}$	**1**(0)$^{\star 4}$	**1**(0)$^{\star 4}$	**1**(0)$^{\star 4}$	**1**(0)$^{\star 4}$	15/15
BIPOP-C	7.9(1)	14(2)	20(2)	26(3)	33(3)	45(3)	57(4)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f2	385	386	387	388	390	391	393	15/15
A2 Carp	56(5)	74(6)	93(7)	112(6)	127(7)	162(8)	197(10)	15/15
A2 Zoub	21(2)	25(4)	26(4)	27(4)	28(3)	29(6)	31(5)	15/15
BFGS	**20**(4)	**24**(5)	**26**(4)	**27**(4)	**27**(4)	**28**(3)	**28**(2)	15/15
BIPOP-C	35(7)	40(3)	44(2)	45(3)	47(3)	48(2)	50(2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f3	5066	7626	7635	7637	7643	7646	7651	15/15
A2 Carp	∞	∞	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	∞	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	∞	∞	∞	∞	∞	∞	∞ *1e5*	0/15
BIPOP-C	**12**(6)$^{\star 4}$	∞	∞	∞	∞	∞	∞ *6e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f4	4722	7628	7666	7686	7700	7758	1.4e5	9/15
A2 Carp	∞	∞	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	∞	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	∞	∞	∞	∞	∞	∞	∞ *2e5*	0/15
BIPOP-C	∞	∞	∞	∞	∞	∞	∞ *6e6*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f5	41	41	41	41	41	41	41	15/15
A2 Carp	7.4(2)	8.5(2)	8.6(3)	8.6(2)	8.6(2)	8.6(2)	8.6(2)	15/15
A2 Zoub	8.2(2)	9.4(2)	10(2)	10(2)	10(2)	10(2)	10(3)	15/15
BFGS	**2.4**(0.4)$^{\star 4}$	**2.7**(0.3)$^{\star 4}$	**2.8**(0.3)$^{\star 4}$	**2.8**(0.3)$^{\star 4}$	**2.8**(0.1)$^{\star 4}$	**2.8**(0.8)$^{\star 4}$	**2.8**(0.5)$^{\star 4}$	15/15
BIPOP-C	5.1(0.8)	6.2(0.9)	6.3(1)	6.3(1)	6.3(1)	6.3(1)	6.3(1)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f6	1296	2343	3413	4255	5220	6728	8409	15/15
A2 Carp	10(0.7)	8.9(0.3)	8.5(0.4)	8.8(0.4)	8.8(0.5)	9.2(0.4)	9.3(0.2)	15/15
A2 Zoub	**1.4**(0.2)	1.3(0.4)	1.3(0.3)	1.3(0.3)	1.3(0.4)	1.5(0.4)	1.6(0.3)	15/15
BFGS	3.6(2)	3.5(1)	3.4(0.9)	3.5(0.8)	3.5(1.0)	3.6(0.7)	45(38)	0/15
BIPOP-C	1.5(0.3)	**1.3**(0.2)	**1.2**(0.2)	**1.1**(0.2)	**1.1**(0.1)	**1.2**(0.1)$^{\star 3}$	**1.2**(0.1)$^{\star 3}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f7	1351	4274	9503	16523	16524	16524	16969	15/15
A2 Carp	6.9(1)	52(29)	150(142)	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	1166(1339)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	∞	∞	∞	∞	∞	∞	∞ *2100*	0/15
BIPOP-C	**1**(0.9)	**4.9**(3)	**3.5**(1)	**2.2**(0.3)	**2.2**(0.2)	**2.2**(0.2)	**2.1**(0.2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f8	2039	3871	4040	4148	4219	4371	4484	15/15
A2 Carp	15(1)	12(0.3)	12(7)	14(0.7)	16(0.5)	20(6)	42(33)	1/15
A2 Zoub	5.4(2)	17(2)	18(40)	18(3)	19(38)	21(2)	23(3)	15/15
BFGS	**1.8**(0.4)$^{\star 2}$	**1.2**(0.1)$^{\star 4}$	**1.2**(0.1)$^{\star 4}$	**1.2**(0.1)$^{\star 4}$	**1.2**(0.2)$^{\star 4}$	**1.2**(0.2)$^{\star 4}$	**1.2**(0.2)$^{\star 4}$	15/15
BIPOP-C	4.0(1)	4.0(0.7)	4.3(0.3)	4.5(1)	4.5(0.6)	4.6(1.0)	4.6(0.5)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f9	1716	3102	3277	3379	3455	3594	3727	15/15
A2 Carp	19(2)	22(16)	23(9)	24(9)	26(22)	27(15)	31(15)	13/15
A2 Zoub	10(6)	59(168)	66(153)	81(164)	99(174)	138(60)	174(217)	15/15
BFGS	**2.2**(0.4)$^{\star 4}$	**2.2**(1)$^{\star 4}$	**2.1**(0.9)$^{\star 4}$	**2.1**(1)$^{\star 4}$	**2.0**(0.9)$^{\star 4}$	**2.0**(1)$^{\star 4}$	**1.9**(1)$^{\star 4}$	15/15
BIPOP-C	4.7(1)	5.7(5)	6.0(3)	6.1(3)	6.1(3)	6.1(4)	6.1(0.9)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f10	7413	8661	10735	13641	14920	17073	17476	15/15
A2 Carp	403(465)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	900(1015)	779(1505)	1341(2511)	1060(2020)	973(637)	1746(1728)	∞ *2e6*	0/15
BFGS	**1.0**(0.1)$^{\star 4}$	**1**(0.1)$^{\star 4}$	**1**(0.3)*	**1.1**(0.6)	**1.1**(0.4)	3.1(5)	∞ *1e6*	0/15
BIPOP-C	1.9(0.2)	1.8(0.0)	1.6(0.1)	1.3(0.1)	1.2(0.0)	**1.1**(0.0)	**1.1**(0.0)$^{\star 4}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f11	1002	2228	6278	8586	9762	12285	14831	15/15
A2 Carp	∞	∞	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	∞	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	**1**(0.7)$^{\star 4}$	**1**(0.8)$^{\star 4}$	**1.3**(0.5)$^{\star 2}$	2.6(3)	147(89)	∞	∞ *2e5*	0/15
BIPOP-C	10(0.5)	5.1(0.2)	1.9(0.0)	**1.5**(0.0)	**1.4**(0.0)$^{\star 4}$	**1.2**(0.0)$^{\star 4}$	**1.0**(0.0)$^{\star 4}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f12	1042	1938	2740	3156	4140	12407	13827	15/15
A2 Carp	34(10)	32(11)	32(14)	45(19)	125(233)	233(197)	∞ *2e5*	0/15
A2 Zoub	86(314)	725(1040)	2194(2643)	2646(7688)	3267(1449)	1091(767)	979(2278)	2/15
BFGS	**1.6**(2)$^{\star 2}$	**1.6**(1)	**1.6**(0.8)	**1.7**(0.5)$^{\star 2}$	**1.6**(2)$^{\star 2}$	**1.8**(3)	45(68)	1/15
BIPOP-C	3.0(0.2)	4.0(4)	4.5(4)	4.9(3)	4.5(2)	1.9(0.9)	**2.0**(0.7)$^{\star 2}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f13	652	2021	2751	3507	18749	24455	30201	15/15
A2 Carp	33(0.4)	39(26)	72(37)	89(79)	76(67)	∞	∞ *2e5*	0/15
A2 Zoub	131(488)	248(415)	938(1292)	2517(2175)	1612(2758)	∞	∞ *2e6*	0/15
BFGS	**1.7**(0.3)$^{\star 3}$	**1**(0.0)$^{\star 2}$	**1**(0.0)$^{\star 2}$	**1**(0.1)	23(19)	∞	∞ *5e5*	0/15
BIPOP-C	4.3(4)	2.7(5)	5.1(6)	6.2(4)	**1.5**(0.7)	**2.3**(2)$^{\star 4}$	**3.0**(2)$^{\star 4}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f14	75	239	304	451	932	1648	15661	15/15
A2 Carp	5.4(3)	22(4)	39(2)	48(4)	41(2)	∞	∞ *2e5*	0/15
A2 Zoub	**2.3**(2)	2.4(0.6)	3.3(0.6)	3.8(0.3)	4.4(0.7)	376(470)	∞ *2e6*	0/15
BFGS	2.7(1)	**1.8**(0.6)	**2.0**(0.8)$^{\star 3}$	**1.8**(0.4)$^{\star 4}$	**1.2**(0.2)$^{\star 4}$	**1.1**(0.2)$^{\star 4}$	∞ *2e5*	0/15
BIPOP-C	3.9(0.9)	2.9(0.5)	3.7(0.5)	4.3(0.6)	4.1(0.4)	6.2(0.4)	**1.2**(0.1)$^{\star 4}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f15	30378	1.5e5	3.1e5	3.2e5	3.2e5	4.5e5	4.6e5	15/15
A2 Carp	∞	∞	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	∞	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	∞	∞	∞	∞	∞	∞	∞ *1e5*	0/15
BIPOP-C	**1**(0.4)$^{\star 4}$	**2.0**(1.0)	**1.4**(0.4)	**1.4**(0.5)	**1.4**(0.5)	**1**(0.4)	**1**(0.3)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f16	1384	27265	77015	1.4e5	1.9e5	2.0e5	2.2e5	15/15
A2 Carp	20(2)	14(9)	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	246(399)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	∞	∞	∞	∞	∞	∞	∞ *3e5*	0/15
BIPOP-C	**1.7**(0.5)	**1.0**(0.6)$^{\star 3}$	**1.2**(1.0)$^{\star 4}$	**1**(0.6)$^{\star 4}$	**1**(0.9)$^{\star 4}$	**1**(0.5)$^{\star 4}$	**1**(0.7)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f17	63	1030	4005	12242	30677	56288	80472	15/15
A2 Carp	2.4(3)	10(2)	7.0(1)	5.0(4)	9.1(7)	∞	∞ *2e5*	0/15
A2 Zoub	**1.5**(1)	3.2e4(3e4)	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	359(591)	∞	∞	∞	∞	∞	∞ *4e5*	0/15
BIPOP-C	2.2(2)	**1**(0.1)$^{\star 4}$	**1**(0.8)$^{\star 4}$	**1**(0.3)$^{\star 4}$	**1.2**(0.8)$^{\star 3}$	**1.3**(0.8)$^{\star 4}$	**1.4**(0.6)$^{\star 4}$	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f18	621	3972	19561	28555	67569	1.3e5	1.5e5	15/15
A2 Carp	7.4(2)	5.8(0.9)	4.0(3)	32(42)	44(54)	∞	∞ *2e5*	0/15
A2 Zoub	622(1925)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	∞	∞	∞	∞	∞	∞	∞ *4e5*	0/15
BIPOP-C	**1.0**(0.4)$^{\star 2}$	**2.4**(2)$^{\star 2}$	**1.2**(1)$^{\star 2}$	**1.6**(2)$^{\star 3}$	**1.1**(0.6)$^{\star 4}$	**1.7**(0.5)	**1.6**(0.5)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f19	1	1	3.4e5	4.7e6	6.2e6	6.7e6	6.7e6	15/15
A2 Carp	**1**(0)	**1**(0)	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	**1**(0)	**1**(0)	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	1.2e6(1e6)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
BIPOP-C	169(40)	2.4e4(3e4)	**1.2**(1)	**1**(0.3)	**1**(0.3)	**1**(0.2)	**1**(0.2)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f20	82	46150	3.1e6	5.5e6	5.5e6	5.6e6	5.6e6	14/15
A2 Carp	28(11)	**2.0**(2)	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	3.7(0.8)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	**2.1**(0.4)$^{\star 3}$	5.8(4)	∞	∞	∞	∞	∞ *4e5*	0/15
BIPOP-C	4.3(0.9)	9.2(2)	**1**(0.0)	**1**(0.9)	**1**(0.3)	**1**(0.5)	**1**(0.3)	14/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f21	561	6541	14103	14318	14643	15567	17589	15/15
A2 Carp	43(3)	49(88)	23(34)	23(12)	23(36)	22(26)	19(28)	6/15
A2 Zoub	457(1020)	388(701)	368(393)	362(651)	354(441)	333(517)	295(405)	5/15
BFGS	**1.9**(3)	**5.5**(6)	**4.6**(5)	**4.6**(2)	**4.5**(5)	**4.3**(3)	**7.3**(9)	2/15
BIPOP-C	3.2(0.5)	55(25)	48(23)	47(58)	46(93)	43(86)	39(54)	13/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f22	467	5580	23491	24163	24948	26847	1.3e5	12/15
A2 Carp	67(216)	151(188)	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	495(1348)	633(930)	719(596)	699(633)	677(694)	630(538)	125(221)	2/15
BFGS	**2.5**(1)	**1.8**(4)	**8.1**(11)	**7.9**(4)	**7.7**(14)	**10**(6)	**14**(11)	0/15
BIPOP-C	6.8(14)	13(21)	215(276)	209(326)	202(240)	188(273)	37(35)	5/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f23	3.2	1614	67457	3.7e5	4.9e5	8.1e5	8.4e5	15/15
A2 Carp	2.0(3)	∞	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	**1.9**(1)	312(449)	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	47(16)	304(261)	∞	∞	∞	∞	∞ *1e5*	0/15
BIPOP-C	4.3(4)	**32**(24)	**1**(0.9)	**1.7**(0.7)	**2.0**(1)	**1.2**(0.8)	**1.2**(0.7)	15/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f24	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	5.2e7	3/15
A2 Carp	∞	∞	∞	∞	∞	∞	∞ *2e5*	0/15
A2 Zoub	∞	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BFGS	∞	∞	∞	∞	∞	∞	∞ *1e5*	0/15
BIPOP-C	**1**(1)	**1**(0.9)	**1**(0.8)	**1**(2)	**1**(1)	**1**(0.6)	**1**(1)	3/15

Table 2: Average running time (aRT in number of function evaluations) divided by the respective best aRT measured during BBOB-2009 in dimension 20. The aRT and in braces, as dispersion measure, the half difference between 10 and 90%-tile of bootstrapped run lengths appear for each algorithm and target, the corresponding best aRT in the first row. The different target Δf-values are shown in the top row. #succ is the number of trials that reached the (final) target $f_{\mathrm{opt}} + 10^{-8}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction of 110. A ↓ indicates the same tested against the best algorithm of BBOB-2009. Best results are printed in bold.