Benchmarking CMAES-APOP on the BBOB Noiseless Testbed

Duc Manh Nguyen
Department of Mathematics and Informatics
Hanoi National University of Education
136 Xuan Thuy, Cau Giay, Hanoi, Vietnam
nguyendm@hnue.edu.vn

Nikolaus Hansen
Inria
Research Centre Saclay, Ecole polytechnique, CMAP
91128 Palaiseau, France
nikolaus.hansen@inria.fr

ABSTRACT

In this paper, we investigate a new approach for adapting population size in the CMA-ES. This method is based on tracking the information in each slot of S successive iterations to decide whether we should increase or decrease or keep the population size in the next slot of S iterations. The information which we collect is the non-decrease of the median of the objective function values. We will show the efficiency of our approach on some multi-modal functions with adequate global structure.

CCS CONCEPTS

•Computing methodologies → Continuous space search;

KEYWORDS

Benchmarking, Black-box optimization, Evolutionary computation, CMA-ES, Popsize Adaptation

ACM Reference format:
Duc Manh Nguyen and Nikolaus Hansen. 2017. Benchmarking CMAES-APOP on the BBOB Noiseless Testbed. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3067695.3084207

1 INTRODUCTION

For multi-modal functions, the default value of population size in the CMA-ES, say $\lambda = \lfloor 4 + 3\log(n) \rfloor$, is known to be insufficient and we need to use a larger population size to get better performance. In [9], Hansen and Kern empirically investigated the effect of the population size on the global search performance of the CMA-ES. They showed that increasing the population size remarkably improves the performance on six of the eight multi-modal functions which have a high number of local optima. For adapting population size, in the literature there are well-known and successful strategies, such as IPOP-CMA-ES [2] in which the CMA-ES is restarted with increasing population by a factor of two whenever one of the stopping criteria is met; and its complicated advancement, BIPOP-CMA-ES strategy [4], which appends a multistart regime with rather a small population size to IPOP-CMA-ES. Besides, Ahrari and Shariat-Panahi [1] proposed a population size adaption method for the CMA-ES which is based on a measure, the oscillation of objective value of x_{mean}, to quantify multimodality of the region under exploration. This quantity is iteratively updated based on the optimization history and subsequently utilized to increase the population size when passing multi-modal regions and vice versa. Recently, Nishida and Akimoto [11] have developed another population size adaptation strategy for the CMA-ES that is based on the estimation accuracy of the natural gradient. This method comes from the observation that the behavior of the CMA-ES on a noisy function is similar to the behavior on the noiseless counterpart except that the number of function evaluations per candidate solution increases.

In this work, we introduce a new approach for adapting population size for the CMA-ES. This approach is inspired from a natural desire when solving an optimization problem as well as one prospect when using larger population size to search: *we want to see the decrease of objective function.* In this method, we will track the non-decrease of objective function in a slot of S successive iterations to adapt population size for the next S successive iterations. It means that we do not adapt population size, say λ, in each iteration but in each slot of S iterations. Consequently, the variation of population size will take a staircase form in iterations.

2 THE CMAES-APOP ALGORITHM

We introduce at first some notations used in this paper:

- k_n : the factor for setting initial population size. It depends on the problem dimension and is quite large to prevent a premature convergence. For instance, it will be set to 10, 20, 30, 40, 50, 60 for $n = 2, 3, 5, 10, 20, 40$ respectively when testing the algorithm on the BBOB noiseless functions.
- iter : number of iterations.
- S : number of iterations in each slot.
- $f^{\text{med}} := \text{median}(f(\mathbf{x}_{i:\lambda}), i = 1, ..., \mu)$: the median of objective function of μ elite solutions in each iteration; $f^{\text{med}}_{\text{prev}}$ and $f^{\text{med}}_{\text{cur}}$ denote the medians in the previous and current iteration respectively.
- n_{up} : the number of times "$f^{\text{med}}_{\text{cur}} - f^{\text{med}}_{\text{prev}} > 0$" occurs during a slot of S iterations.
- t_{up} : the history of n_{up} in each slot recorded.
- no_{up} : the number of most recent slots we do not see the non-decrease.

For solving multi-modal optimization problems, an expectation when increasing population size in the CMA-ES is that we want to see the decrease of objective function. Since with large population size, the algorithm can efficiently explore the search zone to get better solutions. Our algorithm below will track the non-decrease of the objective function in a slot of S successive iterations to adapt population size for next S iterations. The adaptation strategy for

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3084207

population size is based on a natural signal tracked in S iterations: the number of times the f^{med} goes up, say n_{up}, during S successive iterations. This approach allows us to construct a step function of the population size in iteration.

Algorithm: CMAES with an adaptation pop-size strategy

(1) **Input:** $\mathbf{m} \in \mathbb{R}^n, \sigma \in \mathbb{R}_+$
(2) **Initialize:** $\mathbf{C} = \mathbf{I}, \mathbf{p}_c = 0, \mathbf{p}_\sigma = 0, \lambda = k_n \times \lambda_{\text{default}}$
(3) **Set:** $\mu = \lfloor \lambda/2 \rfloor, w_i = \log(\mu + 0.5) - \log i, i = 1, \ldots, \mu,$
$\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i^2}, c_c = \frac{4}{n+4}, c_\sigma = \frac{\mu_w+2}{n+\mu_w+3},$
$c_1 = \frac{2}{(n+1.3)^2+\mu_w}, c_\mu = \frac{2(\mu_w-2+\frac{1}{\mu_w})}{(n+2)^2+\mu_w}, c_1 + c_\mu \le 1,$
$d_\sigma = 1 + 2\max\left(0, \sqrt{\frac{\mu_w-1}{n+1}} - 1\right) + c_\sigma,$
iter $= 0, S = 5, r_{\max} = 30, n_{\text{up}} = 0, t_{\text{up}} = [\ ].$
(4) **While not terminate**
(5) iter = iter + 1;
(6) $\mathbf{x_i} = \mathbf{m} + \sigma \mathbf{y_i}, \mathbf{y_i} \sim \mathbf{N}(\mathbf{0}, \mathbf{C})$, for $i = 1, \ldots, \lambda$
(7) **if** iter = 1
(8) $f^{\text{med}}_{\text{prev}} \leftarrow \text{median}(f(\mathbf{x}_{i:\lambda}), i = 1, \ldots, \mu)$
(9) **else**
(10) $f^{\text{med}}_{\text{cur}} \leftarrow \text{median}(f(\mathbf{x}_{i:\lambda}), i = 1, \ldots, \mu)$
(11) **if** $f^{\text{med}}_{\text{cur}} - f^{\text{med}}_{\text{prev}} > 0$ //Check if f^{med} goes up
(12) $n_{\text{up}} = n_{\text{up}} + 1;$
(13) **end**
(14) **end**
(15) $f^{\text{med}}_{\text{prev}} \leftarrow f^{\text{med}}_{\text{cur}}$ //Reset $f^{\text{med}}_{\text{prev}}$
(16) $\mathbf{m} \leftarrow \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda} = \mathbf{m} + \sigma \mathbf{y}_w$, where $\mathbf{y}_w = \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}$
(17) $\mathbf{p}_c \leftarrow (1-c_c)\mathbf{p}_c + \mathbf{1}_{\|\mathbf{p}_\sigma < 1.5\sqrt{n}\|} \sqrt{(1-(1-c_c)^2)}\sqrt{\mu_w}\mathbf{y}_w$
(18) $\mathbf{p}_\sigma \leftarrow (1-c_\sigma)\mathbf{p}_\sigma + \sqrt{(1-(1-c_\sigma)^2)}\sqrt{\mu_w}\mathbf{C}^{-\frac{1}{2}}\mathbf{y}_w$
(19) $\mathbf{C} \leftarrow (1 - c_1 - c_\mu)\mathbf{C} + c_1 \mathbf{p}_c \mathbf{p}_c^T + c_\mu \sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda} \mathbf{y}_{i:\lambda}^T$
(20) $\sigma \leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma\|}{E\|N(0,1)\|} - 1\right)\right)$
(21) **if** (mod(iter, S) = 1) & (iter > 1) // Adapting pop-size
(22) $t_{\text{up}} = [t_{\text{up}}; n_{\text{up}}];$ // History of n_{up}
(23) **if** $n_{\text{up}} > 1$
(24) $\lambda \leftarrow \left\lfloor \min\left(\exp\left(\frac{n_{\text{up}} \cdot (4+3\log(n))}{S \cdot \sqrt{\lambda - \lambda_{\text{default}} + 1}}\right), r_{\max}\right) \times \lambda \right\rfloor;$
(25) $\lambda \leftarrow \min(\lambda, 400 \times \lambda_{\text{default}})$
(26) $\sigma \leftarrow \sigma \times \exp\left(\frac{1}{n}\left(\frac{n_{\text{up}}}{S} - \frac{1}{5}\right)\right)$; // Enlarge σ a little bit
(27) **elseif** $n_{\text{up}} = 0$
(28) $\text{no}_{\text{up}} = \text{length}(t_{\text{up}}) - \max(\text{find}(t_{\text{up}} > 0));$
(29) **if** $\lambda > 2\lambda_{\text{default}}$
(30) $\lambda \leftarrow \max(\lfloor \lambda \times \exp(-\text{no}_{\text{up}}/10)) \rfloor, 2\lambda_{\text{default}});$
(31) **end**
(32) **end**
(33) **if** λ is changed // Only when $n_{\text{up}} > 1$ or $n_{\text{up}} = 0$
(34) Update $\mu, w_{i=1\ldots\mu}, \mu_w$ w.r.t the new pop-size λ
(35) Update the parameters $c_c, c_\sigma, c_1, c_\mu, d_\sigma$
(36) **end**
(37) $n_{\text{up}} \leftarrow 0$ // Reset n_{up} back to 0
(38) **end**

The lines 7-14 in the algorithm collect information of n_{up} during $S = 5$ iterations in the CMA-ES. After each S iterations, the lines 21-38 adapt the population size based on the information of n_{up} and its history for next S iterations. The algorithm will increase population size if $n_{\text{up}} > 1$, and decrease population size if $n_{\text{up}} = 0$ (there is no "going up" during S iterations) while the population size is not changed if $n_{\text{up}} = 1$. Here, we require threshold 1 for n_{up} to decide about increasing the population size because beside the roughness of the objective function, the (randomly) sampling process could also affect on the information of n_{up}. Let's talk about the formula to increase the population size in line 24:

$$\lambda \leftarrow \left\lfloor \min\left(\exp\left(\frac{n_{\text{up}} \cdot (4 + 3\log(n))}{S \cdot \sqrt{\lambda - \lambda_{\text{default}} + 1}}\right), r_{\max}\right) \times \lambda \right\rfloor.$$

While the parameter $r_{\max} = 30$ is the upper bound of growth rate to avoid increasing too rapidly the population size, here we focus on the factor $\exp\left(\frac{n_{\text{up}} \cdot (4+3\log(n))}{S \cdot \sqrt{\lambda - \lambda_{\text{default}} + 1}}\right)$:

- It suggests a positive effect of n_{up} on the increasing population size, that is the larger the value of n_{up} is (e.g., the search area is very rugged), the more the population size should be increased for next iterations.
- This factor also describes the positive effect of the dimension n (from an intuition that, the larger the dimension of problem is, the harder the multi-modal optimization problem becomes, therefore we need more population), and the negative effect of the current population size on the increase population size for the next S iterations.
  Note that once the population size is increased, reasonably we should increase the step size σ a little bit (line 26 in the algorithm) such that the larger population size could be used more efficiently for the next iteration. The population size is limited to $400 \times \lambda_{\text{default}}$ (line 25).

When there is no going up of f^{med} during a slot of S iterations (i.e., $n_{\text{up}} = 0$), then we will decrease the population size. We first track back the history of n_{up} in the previous slots which is stored in the sequence t_{up} to determine how many times "no going up", say no_{up}, appears in the most recent slots (line 28). The variable no_{up} will decide the speed of population decline by the formula:

$$\lambda \leftarrow \max(\lfloor \lambda \times \exp(-\text{no}_{\text{up}}/10) \rfloor, 2\lambda_{\text{default}}).$$

It reduces about 90% of the current population size if $\text{no}_{\text{up}} = 1$, and about 82% of the current population size if $\text{no}_{\text{up}} = 2$, and so on.

Remark: The median which we use to measure the non-decrease signal makes our algorithm invariant to scaling and shifting operator on the objective function.

3 EXPERIMENTAL PROCEDURE

We test the algorithms with a budget of $10^5 \times n$, where n is the problem dimension, on the BBOB noiseless functions in six different dimensions. We used the matlab implementation of CMA-ES, version 3.40.beta to make CMAES-APOP. It uses the restart strategies as in BIPOP-CMA-ES [4], but without the stagnation condition as follows:

`MaxIter` $= 100 + 50(n+3)^2/\sqrt{\lambda}$ is the maximal number of iterations in each run of CMA-ES.

`TolHistFun` $= 10^{-12}$: the range of the best function values during the last $10 + \lceil 30n/\lambda \rceil$ iterations is smaller than `TolHistFun`.

`EqualFunVals` : in more than $1/3^{\text{rd}}$ of the last n iterations the objective function value of the best and the k-th best solution are identical, that is $f(\mathbf{x}_{i:\lambda}) = f(\mathbf{x}_{k:\lambda})$, where $k = 1 + \lceil 0.1 + \lambda/4 \rceil$.

`TolX` = 10^{-12} : all components of $\mathbf{p}_c$ and all square roots of diagonal components of $\mathbf{C}$, multiplied by σ/σ_0, are smaller than `TolX`.

`TolUpSigma` = 10^{20} : $\sigma/\sigma_0 >$ `TolUpSigma` $\times \sqrt{l}$, where l is the largest eigenvalue of $\mathbf{C}$, indicates a mismatch between σ increase and decrease of all eigenvalues in $\mathbf{C}$. In this, rather untypical, case the progression of the strategy is usually very low and a restart is indicated.

`ConditionCov`: the condition number of $\mathbf{C}$ exceeds 10^{14}.

`NoEffectAxis`: $\mathbf{C}$ remains numerically constant when adding $0.1\sigma\sqrt{l}\mathbf{v}$, where l is the $1 + (\text{iter } mod\ n)$-largest eigenvalue of $\mathbf{C}$ and $\mathbf{v}$ is the corresponding normalized eigenvector.

`NoEffectCoor`: any element of $\mathbf{m}$ remains numerically constant when adding $0.2\sigma l$, where elements of l are the square root of the diagonal elements of $\mathbf{C}$.

The experiment is tested on a MacBook Air Intel(R) Core(TM) i5-5250U CPU @ 1.60GHz, RAM 8G using MATLAB R2013b. Since our algorithm is mainly developed to cope with multi-modal functions therefore in the first run, we use the pure CMA-ES with the default population size $\lambda = \lambda_{\text{default}}$. This pop-size adaptation strategy is applied with initial population size $\lambda = k_n \times \lambda_{\text{default}}$ whenever the algorithm is restarted and then repeated until the budget is used up. From the information of global optimum given in [8], we choose the starting point $\mathbf{m}^0 = \mathbf{0}$ for the first run (using the pure CMA-ES) and second run (the population size adaptation strategy is applied for the first time), and after that $\mathbf{m}^0$ is chosen uniformly in $[-4, 4]^n$. We set the initial step-size $\sigma_0 = 2$ for all run.

We will compare our algorithm with the existing methods IPOP-CMA-ES and BIPOP-CMA-ES which also use pop-size adaptation strategy.

4 PARAMETER TUNING

Because the larger the dimension of problem is, the more the initial population size is needed to prevent a premature convergence. Therefore the parameter k_n, as a function of dimension, will be tuned to 10, 20, 30, 40, 50, 60 for dimension $n = 2, 3, 5, 10, 20, 40$ respectively.

Whenever the population size λ is updated, the parameters corresponding is also changed as follows:

$$\mu = \lfloor \lambda/2 \rfloor, w_i = \log(\mu + 0.5) - \log i, i = 1, ..., \mu,$$

$$\mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i^2}, c_c = \frac{4}{n+4}, c_\sigma = \frac{\mu_w + 2}{n + \mu_w + 3},$$

$$c_1 = \frac{2}{(n+1.3)^2 + \mu_w}, c_\mu = \min\left(1 - c_1, \frac{2(\mu_w - 2 + \frac{1}{\mu_w})}{(n+2)^2 + \mu_w}\right),$$

$$d_\sigma = 1 + 2\max\left(0, \sqrt{\frac{\mu_w - 1}{n+1}} - 1\right) + c_\sigma.$$

5 CPU TIMING

Besides studying the performance of the algorithm, we are interested in evaluating the CPU timing of the algorithm. Thus we run the CMAES-APOP on the BBOB test suite [8] with a small budget. Here, we set maximum budget equal to $400(n + 2)$ function evaluations according to [10]. The time per function evaluation for dimensions 2, 3, 5, 10, 20, 40 equals 2.010269e-06, 1.795326e-06, 1.312460e-06, 7.360946e-07, 8.145508e-07, and 5.276580e-07 seconds respectively.

6 RESULTS

Results from experiments according to [10] and [5] on the benchmark functions given in [3, 8] are presented in Figures 1, 2 and 3 and in Tables 1 and 2. The experiments were performed with COCO [7], version 2.1, the plots were produced with version 2.1.

The **average runtime (aRT)**, used in the figures and tables, depends on a given target function value, $f_t = f_{\text{opt}} + \Delta f$, 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 [6, 12]. **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 and Tables 1 and 2 show that in 5-D (respectively 20-D), CMAES-APOP solves 21 (respectively 19) among 24 functions. This algorithm does not work well for functions $f_{21} - f_{24}$ in the class of multi-modal functions with weak global structure. The unique function which CMAES-APOP works well in this class is function f_{20} (the Schwefel function). However, it shows a good performance on the multi-modal functions with adequate global structure such as: $f_{15}, f_{17}, f_{18}, f_{19}$. It provided a significant improvement of *aRT* for these functions, especially in the case of high dimensions: f_{15} (in dimensions 5, 10, 20 and 40), f_{18} (in dimensions 20 and 40), f_{19} (in dimensions 5, 10), and f_{20} (in dimensions 5, 10 and 20). It also works well on the function f_7 (Step-ellipsoid function) in dimensions 20 and 40. Moreover, a hard function f_3 in the dimension 10 also can be solved efficiently using this pop-size adaptation strategy than with the other methods.

The most important observation that can be made from Figure 1 and Table 2 is that our algorithm runs faster (statistically significant) about 2.5 times than the IPOP-CMA-ES does, and 3-4 times than the BIPOP-CMA-ES does on the multi-modal (but well-structured) function f_{15} in high dimension (say 20-D, 40-D); and faster about 1.8 times than the IPOP-CMA-ES does, and 2.6 times than the BIPOP-CMA-ES does on the function f_{18} in dimension 20. Also, it runs faster about 1.8 times than the IPOP-CMA-ES does, and 2.5 times than the BIPOP-CMA-ES does on the function f_7 in dimension 40. Although the difference has not shown to be statistically significant, it runs faster about 3.5 times than the BIPOP-CMA-ES does on the function f_{19} in dimension 40, and about 2 times on the function f_{20} in dimension 20.

7 CONCLUSION

Our strategy on adapting population-size in the CMA-ES has shown a good performance on some multi-modal functions of the BBOB-2017 testbed, especially on the well-structured multi-modal function f_{15} (Rastrigin function). It means that the signal on non-decrease of objective function which we tracked in order to give an adaptation strategy on population size is quite reasonable. However, this

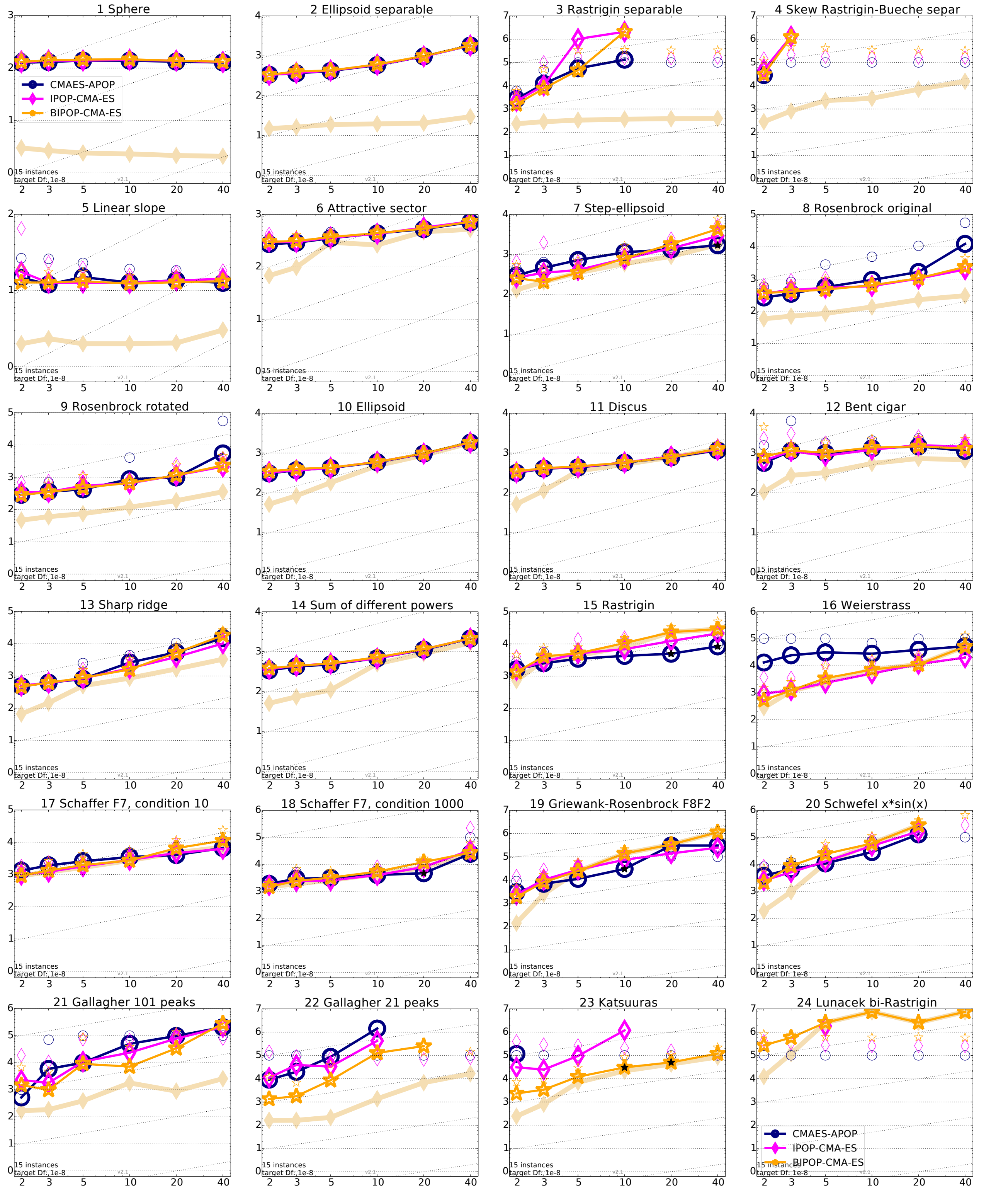

Figure 1: Average running time (aRT in number of f-evaluations as $\log_{10}$ value), divided by dimension for target function value 10^{-8} versus dimension. Slanted grid lines indicate quadratic scaling with the dimension. Different symbols correspond to different algorithms given in the legend of 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: ○: CMAES-APOP, ◇: IPOP-CMA-ES, ⋆: BIPOP-CMA-ES

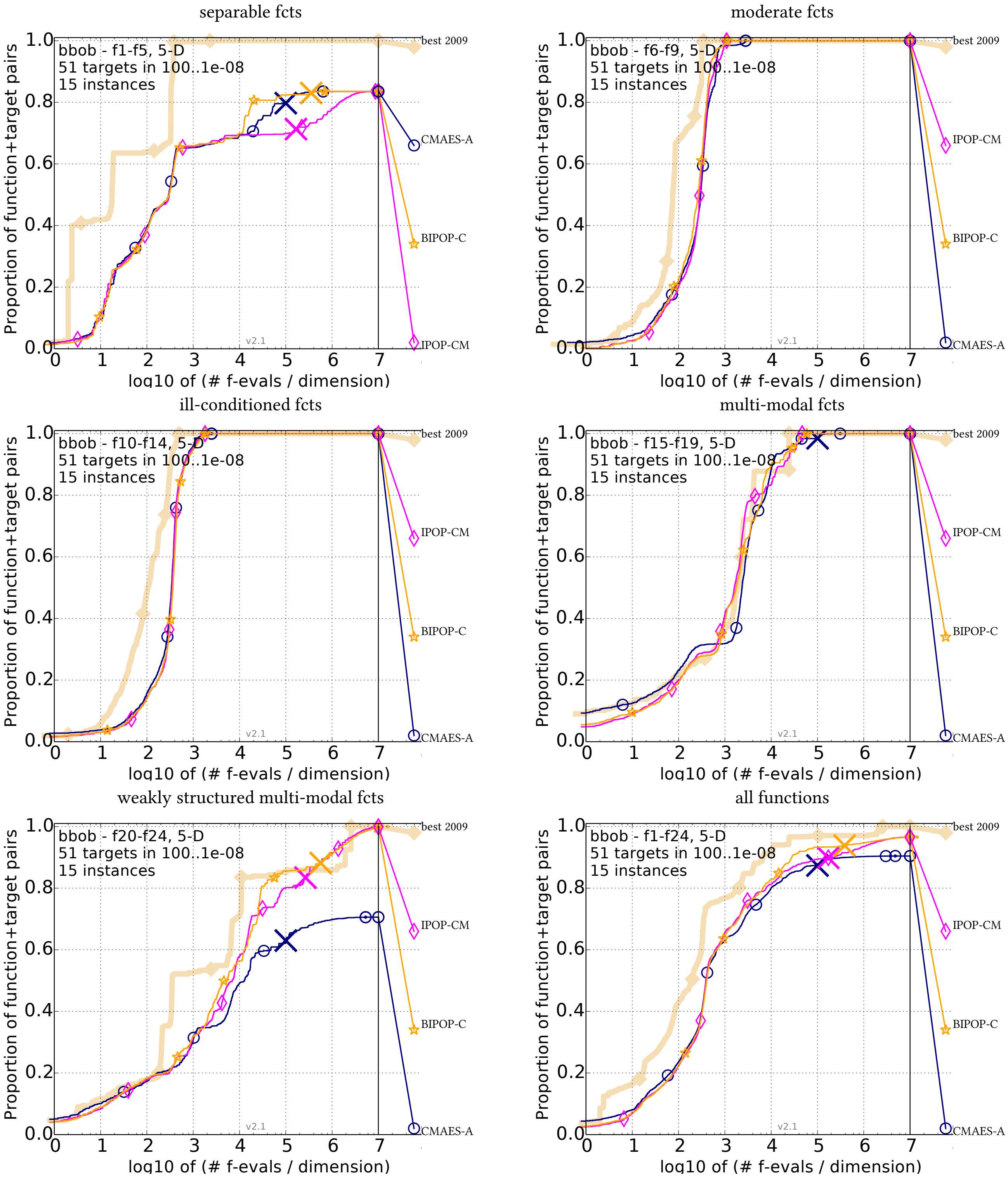

Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 51 targets with target precision in $10^{[-8..2]}$ for all functions and subgroups in 5-D. The "best 2009" line corresponds to the best aRT observed during BBOB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

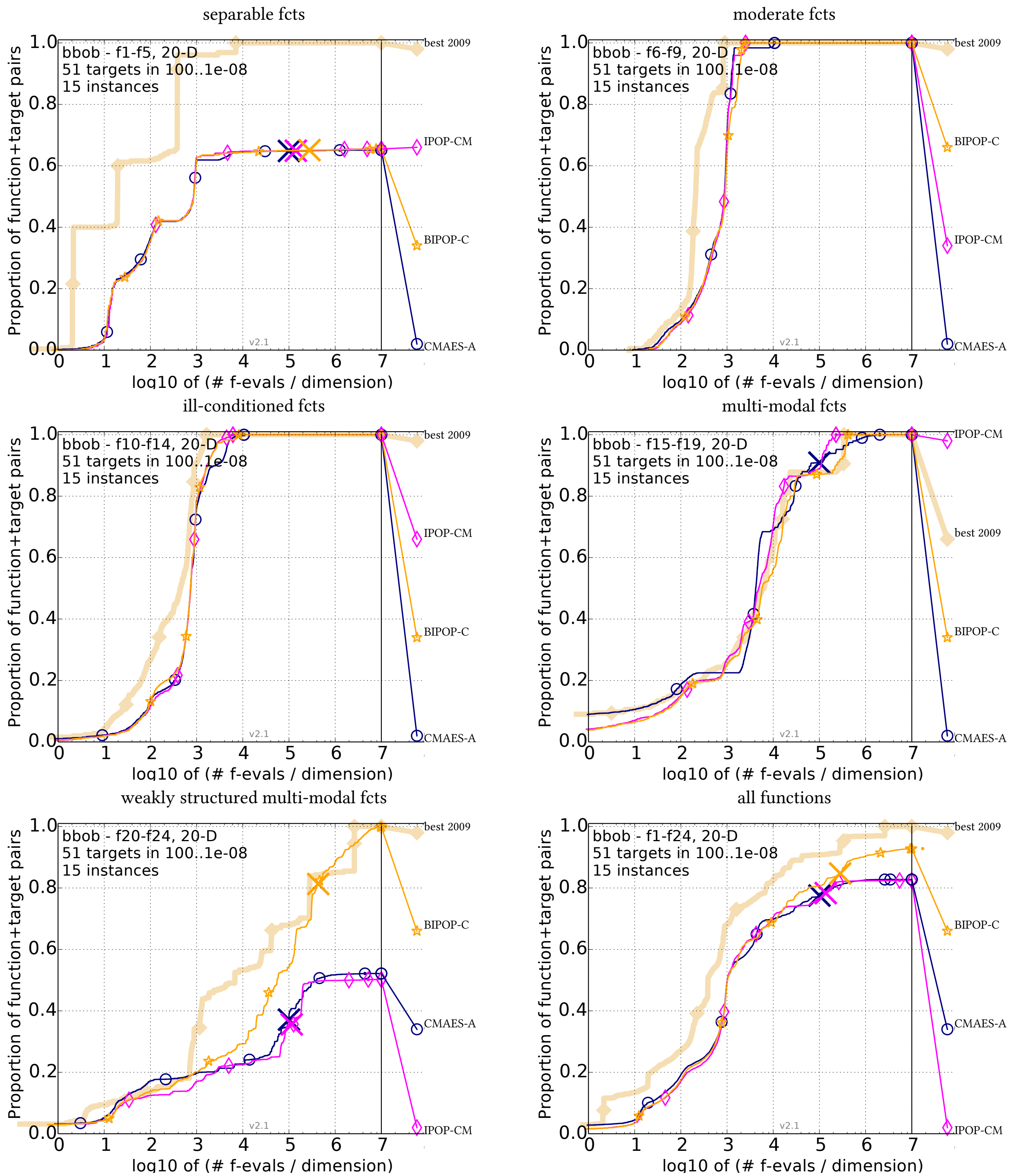

Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 51 targets with target precision in $10^{[-8..2]}$ for all functions and subgroups in 20-D. The "best 2009" line corresponds to the best aRT observed during BBOB 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
CMAES-A	**2.3**(2)	**6.0**(3)	**14**(2)	**19**(3)	**26**(2)	**39**(3)	**52**(4)	15/15
IPOP-CM	2.5(2)	8.1(3)	14(3)	21(3)	28(2)	40(3)	52(3)	15/15
BIPOP-C	3.2(2)	9.1(4)	15(4)	21(4)	28(4)	41(3)	54(6)	15/15
f2	83	87	88	89	90	92	94	15/15
CMAES-A	14(4)	17(1)	**17**(1)	**18**(1)	**19**(1.0)	**20**(1)	**22**(1)	15/15
IPOP-CM	14(4)	16(3)	18(2)	19(1)	19(1)	21(1)	22(2)	15/15
BIPOP-C	**13**(2)	**16**(3)	18(2)	19(2)	20(2)	21(3)	22(2)	15/15
f3	716	1622	1637	1642	1646	1650	1654	15/15
CMAES-A	**1.3**(3)	40(45)	164(254)	166(221)	167(228)	167(218)	168(203)	12/15
IPOP-CM	2.2(1)	70(267)	3130(3338)	3121(5232)	3113(2524)	3106(2874)	3099(3415)	2/15
BIPOP-C	1.4(1)	**16**(21)	**139**(95)	**139**(69)	**139**(518)	**139**(517)	**140**(552)	14/15
f4	809	1633	1688	1758	1817	1886	1903	15/15
CMAES-A	4.1(4)	∞	∞	∞	∞	∞	∞ *5e5*	0/15
IPOP-CM	**2.0**(2)	∞	∞	∞	∞	∞	∞ *9e5*	0/15
BIPOP-C	2.7(3)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
f5	10	10	10	10	10	10	10	15/15
CMAES-A	5.8(3)	7.3(4)	7.4(4)	7.5(3)	7.5(3)	7.5(4)	7.5(3)	15/15
IPOP-CM	4.6(2)	**6.0**(2)	**6.3**(1)	**6.3**(2)	**6.3**(3)	**6.3**(2)	**6.3**(2)	15/15
BIPOP-C	**4.5**(2)	6.5(2)	6.6(3)	6.6(3)	6.6(3)	6.6(2)	6.6(2)	15/15
f6	114	214	281	404	580	1038	1332	15/15
CMAES-A	**1.7**(0.9)	**1.8**(0.5)	**2.0**(0.5)	**1.9**(0.3)	**1.6**(0.3)	**1.2**(0.2)	**1.2**(0.2)	15/15
IPOP-CM	2.5(1)	2.1(0.4)	2.2(0.4)	2.0(0.5)	1.7(0.2)	1.3(0.3)	1.2(0.1)	15/15
BIPOP-C	2.3(1.0)	2.1(0.6)	2.2(0.6)	1.9(0.4)	1.7(0.2)	1.3(0.3)	1.3(0.2)	15/15
f7	24	324	1171	1451	1572	1572	1597	15/15
CMAES-A	**3.2**(2)	2.2(3)	1.7(0.3)	1.9(0.7)	1.9(0.6)	1.9(0.6)	2.0(0.3)	15/15
IPOP-CM	4.4(2)	1.7(2)	1.2(0.8)	1.1(0.8)	1.2(0.5)	1.2(0.5)	1.2(0.6)	15/15
BIPOP-C	4.9(8)	**1.5**(1)	**1.0**(1)	**1.00**(0.9)	**1.0**(0.6)	**1.0**(0.9)	**1.00**(0.5)	15/15
f8	73	273	336	372	391	410	422	15/15
CMAES-A	**2.6**(0.5)	4.4(7)	5.4(0.7)	5.6(7)	5.8(7)	6.1(8)	6.4(14)	15/15
IPOP-CM	3.5(2)	4.8(5)	5.3(0.7)	5.5(5)	5.6(2)	5.8(3)	6.1(0.9)	15/15
BIPOP-C	3.2(2)	**3.7**(0.5)	**4.5**(4)	**4.7**(1)	**4.8**(2)	**5.1**(1)	**5.4**(4)	15/15
f9	35	127	214	263	300	335	369	15/15
CMAES-A	**4.9**(1)	**6.6**(2)	**6.2**(2)	**5.9**(1)	**5.6**(1)	**5.6**(0.9)	**5.5**(0.7)	15/15
IPOP-CM	5.9(1)	11(10)	8.7(6)	7.9(6)	7.5(3)	7.4(4)	7.2(4)	15/15
BIPOP-C	5.8(2)	8.7(3)	7.2(3)	6.7(2)	6.4(4)	6.3(1)	6.2(1)	15/15
f10	349	500	574	607	626	829	880	15/15
CMAES-A	**3.1**(0.7)	**2.7**(0.6)	**2.7**(0.5)	**2.7**(0.5)	**2.7**(0.3)	**2.2**(0.2)	**2.3**(0.2)	15/15
IPOP-CM	3.7(0.6)	2.9(0.4)	2.7(0.4)	2.7(0.2)	2.8(0.3)	2.3(0.2)	2.3(0.2)	15/15
BIPOP-C	3.5(0.7)	2.9(0.4)	2.7(0.4)	2.7(0.3)	2.8(0.3)	2.3(0.2)	2.4(0.1)	15/15
f11	143	202	763	977	1177	1467	1673	15/15
CMAES-A	**7.8**(2)	**6.9**(1)	**2.1**(0.2)	**1.7**(0.2)	**1.5**(0.2)	**1.3**(0.1)	**1.2**(0.1)	15/15
IPOP-CM	8.6(2)	7.3(0.7)	2.1(0.2)	1.8(0.2)	1.6(0.2)	1.4(0.1)	1.3(0.1)	15/15
BIPOP-C	8.3(2)	7.1(2)	2.2(0.3)	1.8(0.1)	1.6(0.3)	1.4(0.1)	1.3(0.1)	15/15
f12	108	268	371	413	461	1303	1494	15/15
CMAES-A	**8.0**(8)	**5.1**(3)	**5.9**(4)	**6.3**(4)	6.6(4)	3.1(0.5)	3.2(1)	15/15
IPOP-CM	9.5(4)	6.1(5)	6.2(2)	6.4(4)	**6.4**(5)	**2.8**(2)	**2.8**(1)	15/15
BIPOP-C	11(8)	7.4(7)	7.4(5)	7.5(5)	7.7(4)	3.3(2)	3.3(2)	15/15
f13	132	195	250	319	1310	1752	2255	15/15
CMAES-A	**2.7**(0.5)	**4.4**(3)	5.9(1)	5.4(1)	1.5(0.2)	1.5(0.2)	**1.6**(2)	15/15
IPOP-CM	3.2(4)	4.9(2)	**5.3**(3)	**5.1**(2)	**1.4**(0.5)	1.6(0.2)	1.6(0.4)	15/15
BIPOP-C	3.9(3)	5.4(4)	5.9(3)	5.4(1)	1.6(0.2)	**1.5**(0.2)	1.7(0.2)	15/15
f14	10	41	58	90	139	251	476	15/15
CMAES-A	**0.64**(0.2)	**1.8**(2)	**3.2**(0.6)	**3.5**(0.7)	**4.4**(1)	**5.0**(1)	**4.2**(0.3)	15/15
IPOP-CM	2.1(3)	2.9(1)	3.8(0.8)	4.3(1)	4.7(1)	5.5(1)	4.4(0.4)	15/15
BIPOP-C	1.1(0.8)	2.8(1)	3.7(1)	4.0(0.8)	4.5(1)	5.4(0.9)	4.5(0.3)	15/15
f15	511	9310	19369	19743	20073	20769	21359	14/15
CMAES-A	**1.5**(2)	**1.2**(0.2)	**0.66**(0.1)	**0.70**(0.2)	**0.73**(0.3)	**0.77**(0.2)	**0.80**(0.2)	15/15
IPOP-CM	2.3(2)	1.3(1)	1.2(1)	1.2(0.8)	1.2(0.7)	1.2(1)	1.2(1)	15/15
BIPOP-C	1.6(1)	1.5(1)	1.2(0.6)	1.2(0.7)	1.2(0.5)	1.2(0.4)	1.2(0.4)	15/15
f16	120	612	2662	10163	10449	11644	12095	15/15
CMAES-A	**2.0**(2)	34(88)	34(23)	9.5(16)	14(25)	13(37)	13(30)	13/15
IPOP-CM	2.5(1)	**2.3**(0.4)	**1.7**(2)	**0.55**(0.4)	**0.96**(0.3)	**0.94**(0.5)	**0.95**(1)	15/15
BIPOP-C	3.0(3)	3.6(3)	2.6(1)	1.1(2)	1.3(1.0)	1.4(1)	1.4(1)	15/15
f17	5.0	215	899	2861	3669	6351	7934	15/15
CMAES-A	**1.7**(1)	**0.95**(0.4)	1.3(2)	1.1(1)	1.6(1)	1.4(0.9)	1.6(0.8)	15/15
IPOP-CM	5.0(5)	1.1(0.5)	**0.97**(0.2)	**0.61**(0.5)	**0.77**(0.8)	**0.81**(0.7)	**1.0**(0.6)	15/15
BIPOP-C	3.5(3)	1.00(0.2)	1.0(2)	1.00(1)	1.00(0.8)	1.00(0.5)	1.2(0.4)	15/15
f18	103	378	3968	8451	9280	10905	12469	15/15
CMAES-A	**0.73**(0.6)	**0.80**(0.2)	**0.67**(0.8)	**0.82**(0.5)	**0.99**(0.3)	1.2(0.2)	1.2(0.3)	15/15
IPOP-CM	1.2(1)	2.7(4)	0.87(0.9)	1.1(0.4)	1.0(0.4)	**1.0**(0.3)	**0.99**(0.2)	15/15
BIPOP-C	1.0(0.6)	3.4(9)	1.0(1)	1.0(0.3)	1.0(0.4)	1.2(0.4)	1.3(0.6)	15/15
f19	1	1	242	1.0e5	1.2e5	1.2e5	1.2e5	15/15
CMAES-A	**1**(0)$^{\star 2}$	**1**(0)$^{\star 4}$	**90**(20)	**0.41**(0.2)	**0.40**(0.2)	**0.43**(0.1)	**0.45**(0.2)	15/15
IPOP-CM	21(26)	1720(2062)	125(160)	1.2(0.7)	1.1(0.5)	1.1(0.7)	1.1(0.5)	15/15
BIPOP-C	20(13)	2801(1533)	161(184)	1.00(0.8)	1.00(1.0)	1.0(0.9)	1.00(0.7)	15/15
f20	16	851	38111	51362	54470	54861	55313	14/15
CMAES-A	**1.5**(1.0)	16(6)	**1.1**(0.7)	**0.88**(0.8)	**0.89**(0.7)	**0.93**(0.5)	**0.97**(0.3)	15/15
IPOP-CM	3.9(3)	11(4)	1.4(2)	1.1(1)	1.1(0.5)	1.1(1)	1.1(0.9)	15/15
BIPOP-C	3.3(3)	**8.2**(9)	2.8(2)	2.2(1)	2.1(0.8)	2.2(2)	2.2(0.7)	15/15
f21	41	1157	1674	1692	1705	1729	1757	14/15
CMAES-A	**1.8**(1)	11(1)	27(10)	27(10)	29(74)	28(11)	28(2)	14/15
IPOP-CM	6.3(10)	**5.6**(4)	30(66)	30(35)	31(16)	31(31)	31(75)	14/15
BIPOP-C	2.3(2)	14(8)	**24**(112)	**25**(62)	**25**(4)	**25**(79)	**25**(62)	15/15
f22	71	386	938	980	1008	1040	1068	14/15
CMAES-A	**6.6**(12)	43(91)	253(408)	441(390)	430(785)	418(273)	407(376)	9/15
IPOP-CM	12(0.4)	48(43)	165(188)	165(14)	161(165)	158(322)	155(256)	11/15
BIPOP-C	6.9(13)	**20**(3)	**45**(98)	**43**(69)	**42**(68)	**41**(95)	**40**(92)	15/15
f23	3.0	518	14249	27890	31654	33030	34256	15/15
CMAES-A	2.4(2)	24(64)	493(379)	∞	∞	∞	∞ *5e5*	0/15
IPOP-CM	2.2(2)	26(43)	33(45)	17(35)	15(40)	14(29)	14(29)	11/15
BIPOP-C	**1.7**(2)	**13**(15)	**3.7**(4)	**2.1**(2)	**1.8**(2)	**1.8**(2)	**1.8**(1)	15/15
f24	1622	2.2e5	6.4e6	9.6e6	9.6e6	1.3e7	1.3e7	3/15
CMAES-A	2.4(3)	34(30)	∞	∞	∞	∞	∞ *5e5*	0/15
IPOP-CM	2.9(3)	18(26)	1.4(2)	**0.94**(2)	**0.94**(0.5)	**0.70**(2)	**0.70**(0.8)	2/15
BIPOP-C	**2.1**(3)	**1.6**(3)*	**1**(0.6)	1.0(1)	1.0(1.0)	1(2)	1(1)	3/15

Table 1: Average runtime (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 reference 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 $f_{opt} + 10^{-8}$. 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 from BBOB 2009. Best results are printed in bold.

Data produced with COCO v2.1

method does not work well on the class of multi-modal functions with weak global structure. How to improve this strategy to cope with this class will be our direction in future work.

Acknowledgments

This work was supported by the NumBBO project (grant ANR-2012-MONU-0009) of the French National Research Agency. We would like to thank the BBOB-2017 organizers for their comments on this work.

REFERENCES

[1] A. Ahrari and M. Shariat-Panahi. 2015. An improved evolution strategy with adaptive population size. *Optimization* 64, 12 (2015), 2567–2586.

[2] A. Auger and N. Hansen. 2005. A restart CMA evolution strategy with increasing population size. In *2005 IEEE Congress on Evolutionary Computation*, Vol. 2. 1769–1776 Vol. 2.

[3] S. Finck, N. Hansen, R. Ros, and A. Auger. 2009. *Real-Parameter Black-Box Optimization Benchmarking 2009: Presentation of the Noiseless Functions.* Technical Report 2009/20. Research Center PPE. http://coco.lri.fr/downloads/download15.03/bbobdocfunctions.pdf Updated February 2010.

[4] Nikolaus Hansen. 2009. Benchmarking a BI-population CMA-ES on the BBOB-2009 Function Testbed. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers (GECCO*

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
CMAES-A	**6.3**(1)*	**12**(1)	**18**(2)*	**24**(2)	**31**(2)	**43**(3)	**56**(3)	15/15
IPOP-CM	8.0(1.0)	14(1)	20(3)	26(2)	33(2)	46(2)	58(2)	15/15
BIPOP-C	7.9(2)	14(2)	20(3)	26(3)	33(3)	45(4)	57(3)	15/15
f2	385	386	387	388	390	391	393	15/15
CMAES-A	**33**(3)	**39**(4)	**43**(4)	45(4)	46(3)	47(2)	49(2)	15/15
IPOP-CM	35(4)	41(4)	43(3)	**44**(3)	**45**(2)	**47**(2)	**48**(1)	15/15
BIPOP-C	35(6)	40(4)	44(3)	45(3)	47(2)	48(2)	50(2)	15/15
f3	5066	7626	7635	7637	7643	7646	7651	15/15
CMAES-A	24(11)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
IPOP-CM	13(11)	∞	∞	∞	∞	∞	∞ *3e6*	0/15
BIPOP-C	**12**(6)	∞	∞	∞	∞	∞	∞ *6e6*	0/15
f4	4722	7628	7666	7686	7700	7758	1.4e5	9/15
CMAES-A	∞	∞	∞	∞	∞	∞	∞ *2e6*	0/15
IPOP-CM	∞	∞	∞	∞	∞	∞	∞ *3e6*	0/15
BIPOP-C	∞	∞	∞	∞	∞	∞	∞ *6e6*	0/15
f5	41	41	41	41	41	41	41	15/15
CMAES-A	5.1(0.6)	6.3(0.4)	6.6(0.9)	6.6(0.7)	6.6(0.7)	6.6(2)	6.6(1)	15/15
IPOP-CM	5.7(1)	6.5(1.0)	6.6(0.9)	6.6(1)	6.6(0.7)	6.6(1)	6.6(0.6)	15/15
BIPOP-C	**5.0**(1)	**6.1**(1)	**6.2**(1)	**6.2**(0.6)	**6.3**(1)	**6.3**(1)	**6.3**(1.0)	15/15
f6	1296	2343	3413	4255	5220	6728	8409	15/15
CMAES-A	1.6(0.4)	1.3(0.2)	**1.1**(0.2)	**1.1**(0.2)	**1.1**(0.2)	**1.1**(0.2)	**1.1**(0.2)	15/15
IPOP-CM	1.7(0.2)	1.3(0.2)	1.2(0.1)	1.2(0.1)	1.2(0.1)	1.2(0.1)	1.2(0.1)	15/15
BIPOP-C	**1.5**(0.4)	**1.3**(0.2)	1.2(0.2)	1.1(0.2)	1.1(0.2)	1.2(0.1)	1.2(0.1)	15/15
f7	1351	4274	9503	16523	16524	16524	16969	15/15
CMAES-A	1.4(0.3)	4.9(0.2)	**2.4**(0.1)	**1.6**(0.1)	**1.6**(0.1)	**1.6**(0.1)	**1.5**(0.1)	15/15
IPOP-CM	1.9(2)	**4.8**(1)	2.7(1)	1.7(0.7)	1.7(1.0)	1.7(0.7)	1.6(0.5)	15/15
BIPOP-C	**1.00**(0.2)	4.9(3)	3.5(1)	2.2(0.1)	2.2(0.3)	2.2(0.2)	2.1(0.3)	15/15
f8	2039	3871	4040	4148	4219	4371	4484	15/15
CMAES-A	**3.6**(0.5)	6.5(0.2)	7.0(0.2)	7.1(11)	7.2(11)	7.2(11)	7.3(0.2)	15/15
IPOP-CM	3.7(0.5)	**3.9**(0.3)	**4.2**(0.6)	**4.3**(0.6)	**4.4**(0.4)	**4.4**(0.5)	**4.5**(0.3)	15/15
BIPOP-C	4.0(1)	4.0(1)	4.3(0.5)	4.5(0.9)	4.5(0.5)	4.6(0.6)	4.6(1)	15/15
f9	1716	3102	3277	3379	3455	3594	3727	15/15
CMAES-A	**4.0**(0.9)	**4.6**(0.6)	**5.0**(0.9)	**5.1**(0.4)	**5.2**(0.5)	**5.2**(0.5)	**5.2**(0.6)	15/15
IPOP-CM	4.6(0.4)	5.7(0.4)	6.0(0.4)	6.1(0.4)	6.1(2)	6.1(2)	6.1(2)	15/15
BIPOP-C	4.7(1)	5.7(3)	6.0(0.4)	6.1(4)	6.1(0.9)	6.1(2)	6.1(0.6)	15/15
f10	7413	8661	10735	13641	14920	17073	17476	15/15
CMAES-A	1.8(0.1)	1.8(0.1)	1.6(0.1)	1.3(0.0)	1.2(0.0)	1.1(0.0)	1.1(0.0)	15/15
IPOP-CM	**1.8**(0.3)	**1.8**(0.2)	**1.5**(0.2)	**1.3**(0.1)	**1.2**(0.0)	**1.1**(0.0)	**1.1**(0.1)	15/15
BIPOP-C	1.9(0.2)	1.8(0.2)	1.6(0.1)	1.3(0.1)	1.2(0.0)	1.1(0.0)	1.1(0.0)	15/15
f11	1002	2228	6278	8586	9762	12285	14831	15/15
CMAES-A	**10**(0.7)	**5.0**(0.2)	**1.9**(0.1)	**1.5**(0.1)	**1.4**(0.0)	**1.2**(0.0)	**1.0**(0.0)	15/15
IPOP-CM	11(2)	5.4(1.0)	2.1(0.3)	1.6(0.1)	1.4(0.1)	1.2(0.0)	1.1(0.1)	15/15
BIPOP-C	10(0.5)	5.1(0.3)	1.9(0.1)	1.5(0.0)	1.4(0.0)	1.2(0.0)	1.0(0.0)	15/15
f12	1042	1938	2740	3156	4140	12407	13827	15/15
CMAES-A	**2.6**(0.1)	**3.2**(4)	4.5(3)	**4.9**(2)	**4.4**(2)	**1.9**(0.7)	**2.0**(0.7)	15/15
IPOP-CM	4.8(3)	5.3(5)	5.5(4)	5.9(2)	5.1(4)	2.1(0.8)	2.2(1)	15/15
BIPOP-C	3.0(4)	4.0(2)	**4.5**(3)	4.9(2)	4.5(2)	1.9(0.7)	2.0(0.7)	15/15
f13	652	2021	2751	3507	18749	24455	30201	15/15
CMAES-A	8.3(5)	8.0(8)	8.8(7)	8.8(7)	2.8(2)	3.4(3)	3.3(2)	15/15
IPOP-CM	6.5(6)	4.8(2)	6.2(2)	**5.1**(4)	**1.4**(0.9)	**1.7**(0.6)	**2.3**(1.0)	15/15
BIPOP-C	**4.3**(6)	**2.7**(2)	**5.1**(5)	6.2(3)	1.5(0.8)	2.3(2)	3.0(2)	15/15
f14	75	239	304	451	932	1648	15661	15/15
CMAES-A	**2.6**(1)	**2.5**(0.3)	**3.2**(0.2)*	**4.0**(0.4)	4.0(0.4)	6.0(0.4)	1.2(0.1)	15/15
IPOP-CM	3.7(2)	2.8(0.7)	3.6(0.2)	4.2(0.8)	**3.9**(0.4)	**6.0**(0.5)	**1.2**(0.1)	15/15
BIPOP-C	3.9(1)	2.9(0.4)	3.7(0.2)	4.3(0.6)	4.1(0.3)	6.2(0.6)	1.2(0.1)	15/15
f15	30378	1.5e5	3.1e5	3.2e5	3.2e5	4.5e5	4.6e5	15/15
CMAES-A	2.8(0.4)	**0.61**(0.1)$^{\star 3}$	**0.29**(0.0)$^{\star 3}$	**0.29**(0.0)$^{\star 3}$	**0.29**(0.0)$^{\star 3}$	**0.21**(0.0)$^{\star 3}$	**0.21**(0.0)$^{\star 3}$	15/15
IPOP-CM	1.1(0.7)	1.1(0.4)	0.69(0.3)	0.70(0.4)	0.70(0.3)	0.52(0.2)	0.53(0.2)	15/15
BIPOP-C	**1.0**(0.4)	2.0(1)	1.4(0.3)	1.4(0.5)	1.4(0.4)	1.0(0.3)	1.00(0.3)	15/15
f16	1384	27265	77015	1.4e5	1.9e5	2.0e5	2.2e5	15/15
CMAES-A	22(0.4)	19(15)	8.1(5)	4.5(1)	3.4(2)	3.9(6)	3.5(4)	14/15
IPOP-CM	1.7(1)	**0.81**(0.5)	**0.92**(0.4)	**0.85**(0.4)	**0.84**(0.3)	1.1(0.6)	1.0(0.5)	15/15
BIPOP-C	**1.7**(0.5)	1.0(1.0)	1.2(0.9)	1.0(0.9)	1.00(0.6)	**1.0**(0.8)	**1.00**(0.5)	15/15
f17	63	1030	4005	12242	30677	56288	80472	15/15
CMAES-A	**1.3**(1)	**0.84**(0.1)	1.3(3)	2.5(2)	1.6(0.3)	1.2(0.2)	**0.97**(0.1)	15/15
IPOP-CM	2.1(1)	0.94(0.1)	1.2(2)	1.0(0.4)	**0.76**(0.6)	**0.99**(0.3)	1.0(0.5)	15/15
BIPOP-C	2.2(1)	1.0(0.3)	**1.0**(2)	**1.0**(0.7)	1.2(0.8)	1.3(0.6)	1.4(0.6)	15/15
f18	621	3972	19561	28555	67569	1.3e5	1.5e5	15/15
CMAES-A	**0.88**(0.2)	3.0(5)	1.8(0.7)	1.8(0.2)	**0.85**(0.1)	**0.57**(0.1)$^{\star 3}$	**0.62**(0.2)$^{\star 2}$	15/15
IPOP-CM	1.1(0.5)	**1.8**(2)	**1.1**(0.6)	**1.5**(0.4)	0.97(0.5)	1.0(0.3)	1.1(0.3)	15/15
BIPOP-C	1.0(0.3)	2.4(2)	1.2(0.7)	1.6(1)	1.1(0.5)	1.7(0.6)	1.6(0.3)	15/15
f19	1	1	3.4e5	4.7e6	6.2e6	6.7e6	6.7e6	15/15
CMAES-A	**1**(0)$^{\star 4}$	**1**(0)$^{\star 4}$	0.91(0.1)	**0.40**(0.4)	0.50(0.6)	0.91(1)	0.91(1)	4/15
IPOP-CM	161(51)	2.7e4(1e4)	**0.71**(0.6)	0.45(0.1)	**0.38**(0.1)	**0.41**(0.2)	**0.41**(0.2)	15/15
BIPOP-C	169(38)	2.4e4(2e4)	1.2(0.8)	1.0(0.5)	1.0(0.2)	1.0(0.3)	1(0.3)	15/15
f20	82	46150	3.1e6	5.5e6	5.5e6	5.6e6	5.6e6	14/15
CMAES-A	**3.7**(0.9)	12(5)	**0.34**(0.2)*	**0.46**(0.2)	**0.46**(0.3)	**0.46**(0.4)	**0.46**(0.6)	9/15
IPOP-CM	4.6(1)	**6.4**(2)	0.65(0.2)	0.57(0.2)	0.57(0.2)	0.58(0.2)	0.58(0.1)	15/15
BIPOP-C	4.3(0.9)	9.2(4)	1.00(0.0)	1.00(0.3)	1.00(0.7)	1.00(0.9)	1.00(0.3)	14/15
f21	561	6541	14103	14318	14643	15567	17589	15/15
CMAES-A	4.0(6)	229(314)	138(252)	136(110)	133(239)	125(96)	111(33)	8/15
IPOP-CM	3.7(9)	139(308)	110(297)	108(205)	106(139)	100(104)	88(77)	7/15
BIPOP-C	**3.2**(6)	**55**(124)	**48**(38)	**47**(31)	**46**(58)	**43**(44)	**39**(54)	13/15
f22	467	5580	23491	24163	24948	26847	1.3e5	12/15
CMAES-A	82(445)	2342(1792)	∞	∞	∞	∞	∞ *2e6*	0/15
IPOP-CM	445(1397)	287(346)	∞	∞	∞	∞	∞ *1e6*	0/15
BIPOP-C	**6.8**(13)	**13**(8)	**215**(263)	**209**(293)	**202**(100)	**188**(180)	**37**(94)	5/15
f23	3.0	1614	67457	3.7e5	4.9e5	8.1e5	8.4e5	15/15
CMAES-A	**1.6**(0.8)	279(316)	416(616)	∞	∞	∞	∞ *2e6*	0/15
IPOP-CM	4.6(8)	2.3e4(4e4)	∞	∞	∞	∞	∞ *3e6*	0/15
BIPOP-C	4.6(6)	**32**(31)	**1.00**(0.9)$^{\star 2}$	**1.7**(1)$^{\star 4}$	**2.0**(0.5)$^{\star 4}$	**1.2**(0.9)$^{\star 4}$	**1.2**(0.9)$^{\star 4}$	15/15
f24	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	5.2e7	3/15
CMAES-A	∞	∞	∞	∞	∞	∞	∞ *2e6*	0/15
IPOP-CM	∞	∞	∞	∞	∞	∞	∞ *5e6*	0/15
BIPOP-C	**1.00**(0.9)$^{\star 3}$	**1.00**(1)	**1.0**(0.7)	**1**(0.7)	**1.00**(1)	**1**(1)	**1.00**(0.8)	3/15

Table 2: Average runtime (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 reference 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 $f_{opt} + 10^{-8}$. 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 from BBOB 2009. Best results are printed in bold.

Data produced with COCO v2.1

'09). 2389–2396.

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

[6] N. Hansen, A. Auger, S. Finck, and R. Ros. 2012. *Real-Parameter Black-Box Optimization Benchmarking 2012: Experimental Setup.* Technical Report. INRIA. http://coco.gforge.inria.fr/bbob2012-downloads

[7] N. Hansen, A. Auger, O. Mersmann, T. Tušar, and D. Brockhoff. 2016. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. *ArXiv e-prints* arXiv:1603.08785 (2016).

[8] N. Hansen, S. Finck, R. Ros, and A. Auger. 2009. *Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions.* Technical Report RR-6829. INRIA. http://coco.lri.fr/downloads/download15.03/bbobdocfunctions.pdf Updated February 2010.

[9] N. Hansen and S. Kern. 2004. *Evaluating the CMA Evolution Strategy on Multimodal Test Functions.* 282–291 pages. DOI: http://dx.doi.org/10.1007/978-3-540-30217-9_29

[10] N. Hansen, T. Tušar, O. Mersmann, A. Auger, and D. Brockhoff. 2016. COCO: The Experimental Procedure. *ArXiv e-prints* arXiv:1603.08776 (2016).

[11] K. Nishida and Y. Akimoto. 2016. Population Size Adaptation for the CMA-ES Based on the Estimation Accuracy of the Natural Gradient. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO '16)*. 237–244.

[12] Kenneth Price. 1997. Differential evolution vs. the functions of the second ICEO. In *Proceedings of the IEEE International Congress on Evolutionary Computation.* 153–157.