



HAL
open science

Benchmarking Large Scale Variants of CMA-ES and L-BFGS-B on the bbob-largescale Testbed

Konstantinos Varelas

► **To cite this version:**

Konstantinos Varelas. Benchmarking Large Scale Variants of CMA-ES and L-BFGS-B on the bbob-largescale Testbed. GECCO 2019 Companion - The Genetic and Evolutionary Computation Conference, Jul 2019, Prague, Czech Republic. 10.1145/3319619.3326893 . hal-02160106

HAL Id: hal-02160106

<https://inria.hal.science/hal-02160106v1>

Submitted on 19 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Benchmarking Large Scale Variants of CMA-ES and L-BFGS-B on the bbob-largescale Testbed

Konstantinos Varelas^{1,2}

¹Thales LAS France SAS

²Inria, CMAP, École Polytechnique, IP Paris
firstname.lastname@inria.com

ABSTRACT

In this paper we benchmark five variants of CMA-ES for optimization in large dimension on the novel large scale testbed of COCO under default or modified parameter settings. In particular, we compare the performance of the separable CMA-ES, of VD-CMA-ES and VxD-CMA-ES, of two implementations of the Limited Memory CMA-ES and of the Rank m Evolution Strategy, RmES. For VxD-CMA-ES we perform experiments with different complexity models of the search distribution and for RmES we study the impact of the number of evolution paths employed by the algorithm. The quasi-Newton L-BFGS-B algorithm is also benchmarked and we investigate the effect of choosing the maximum number of variable metric corrections for the Hessian approximation. As baseline comparison, we provide results of CMA-ES up to dimension 320.

CCS CONCEPTS

• **Computing methodologies** → **Continuous space search**;

KEYWORDS

Benchmarking, Black-box optimization, Large scale optimization, CMA-ES, Large scale variants

ACM Reference format:

Konstantinos Varelas^{1,2}. 2019. Benchmarking Large Scale Variants of CMA-ES and L-BFGS-B on the bbob-largescale Testbed. In *Proceedings of Genetic and Evolutionary Computation Conference Companion, Prague, Czech Republic, July 13–17, 2019 (GECCO '19 Companion)*, 9 pages. <https://doi.org/10.1145/3319619.3326893>

1 INTRODUCTION

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [9] is a successful and robust stochastic gradient-free optimizer, addressing difficult non-smooth, non-convex problems. It is a rank-based selection algorithm that adapts a multivariate normal distribution $\mathcal{N}(\mathbf{m}, \sigma^2 \mathbf{C})$ for candidate solution sampling. The quadratic time and space complexity of the method has led researchers to propose several large scale variants over the last years.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6748-6/19/07...\$15.00

<https://doi.org/10.1145/3319619.3326893>

The Comparing Continuous Optimizers platform (COCO) [7] facilitates the automated benchmarking and performance comparison of optimization solvers. In this study, we benchmark promising large scale variants of CMA-ES on the novel bbob-largescale testbed [4] of COCO that extends the bbob suite [8] in dimensions up to 640, maintaining a linear scaling of the function evaluation cost. We include data of CMA-ES up to dimension 320 as baseline and also compare the performance of the gradient based L-BFGS-B method [12] for high dimensional problems, for which we investigate the effect of tuning the parameter that determines the maximum number of directions used for the Hessian approximation.

2 ALGORITHM PRESENTATION AND PARAMETER SETTING

The Python implementation of CMA-ES¹ provides the option of sampling from a distribution with a diagonal covariance matrix. The method, originally introduced as separable CMA-ES [16], defines the model with the simplest complexity, and it is benchmarked here under the default parameter setting, denoted sepCMA in the results.

The VD-CMA-ES [1] and its generalization, VxD-CMA-ES [3] [2], are based on restricting the covariance matrix to models of the forms $\mathbf{D}(\mathbf{I} + \mathbf{v}\mathbf{v}^T)\mathbf{D}$ and $\mathbf{D}(\mathbf{I} + \mathbf{V}\mathbf{V}^T)\mathbf{D}$ respectively, where \mathbf{V} is a matrix composed of k orthogonal column vectors: $\mathbf{V} = [\mathbf{v}_1 \dots \mathbf{v}_k]$. We benchmark the Python implementation of both variants, and in the case of VxD-CMA-ES, we consider fixed $k = 2$, denoted as V2D, as well as online adaptation of k .

The Limited Memory CMA-ES builds on the Cholesky-CMA-ES [17] which conducts only the rank-one update of the covariance matrix implicitly, by operating only on the Cholesky factors of \mathbf{C} . We benchmark the original LMCMA implementation [13], denoted LMCMA14 (14lm in Tables 2 and 3), and a more recent version [14], LMCMA17 (17lm in Tables 2 and 3), both written in C, with the source code obtained from the author² and with their default parameter setting.

The RmES algorithm [11] is maintaining m evolution paths and the resulting search distribution has m principal search directions. The number m of paths is a key parameter for the algorithm. In our comparison, we consider the values $m = 2$, denoted R2ES, and $m = 10$, denoted R10ES. The MATLAB implementation was kindly provided by the authors.

The large scale quasi Newton L-BFGS-B algorithm is also benchmarked with finite difference approximation of the gradient. We benchmark its implementation from the latest version³ of the Python

¹pycma

²publicly available in: LMCMA14 and LMCMA17

³Version 1.2.1

SciPy library. The parameter `maxcor` that controls the maximum number of variable metric corrections used for the Hessian approximation is set to 10 (default value), denoted L-BFGS-B, and to $2 \times D$, D being the dimension, denoted m2DLBFGS. The motivation for this is that experiments on the bbob suite suggest a performance improvement, notably for the Ellipsoid, Discus, Sharp Ridge and Sum of Different Powers functions, using values larger than the default. Such an improvement does not further appear for values larger than $2 \times D$.

Furthermore, results of the python implementation of CMA-ES in its default setting, denoted CMA, are provided for problem dimensions up to 320, as baseline for the performance comparison.

3 EXPERIMENTAL PROCEDURE

We run the algorithms on the entire bbob-largescale suite for $5 \times 10^4 D$ function evaluations according to [10]. A policy of independent restarts is followed when default termination conditions are met. Only for L-BFGS-B, the parameter `ftol` that sets the f tolerance termination condition was changed to the machine precision⁴ for very high accuracy. For all solvers, the initial point was chosen uniformly at random in $[-4, 4]^D$ and for CMA-ES and its variants, the initial step size was set to 2.

4 CPU TIMING

The complete experiment was run on several multicore machines with different processor types and number of cores. In order to evaluate the CPU timing of each algorithm, we have performed a shorter experiment, running the solvers with restarts on the first 3 instances of each function of the bbob-largescale test suite for $100 \times D$ function evaluations. For this, we used (not exclusively) two Linux multicore machines. The time per function evaluation, measured in 10^{-5} seconds for dimensions 20, 40, 80, 160, 320, 640 along with the corresponding processor type and number of cores, are presented in Table 1. The MATLAB implementation of RmES was run with MATLAB R2019a.

5 RESULTS

Results from experiments according to [10] and [5] on the benchmark functions given in [4] are presented in Figures 1, 2, 3, 4 and 5 and Tables 2 and 3. The experiments were performed with COCO [7], version 2.2.1⁵, the plots were produced with version 2.3.3.

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 [6, 15]. **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.

⁴Equal to $2.220446049250313 \times 10^{-16}$

⁵The code that was used was under development, with no difference in the definition of the bbob-largescale testbed, which was officially included in version 2.3 of COCO

Algorithm	Processor Type	20-D	40-D	80-D	160-D	320-D	640-D
sepCMA	Intel Core Haswell, no TSX @2.29 GHz-28 cores	19	23	34	56	100	200
LMCMA17	Intel Core Haswell, no TSX @2.29 GHz-28 cores	4.85	8.70	16.7	38.4	69.0	136
LMCMA14	Intel Core Haswell, no TSX @2.29 GHz-28 cores	5.54	9.40	18.7	38.2	77.9	149
R2ES	Intel(R) Xeon(R) CPU E5-2640 v4 @2.40GHz-40 cores	10.2	13.4	22.2	40.0	79.3	157.4
R10ES	Intel(R) Xeon(R) CPU E5-2640 v4 @2.40GHz-40 cores	9.4	13.2	22.5	39.1	79.8	156.7
VD-CMA	Intel Core Haswell, no TSX @2.29 GHz-28 cores	23	27	37	57	100	200
VkD-CMA	Intel Core Haswell, no TSX @2.29 GHz-28 cores	30	36	45	64	110	200
V2D-CMA	Intel Core Haswell, no TSX @2.29 GHz-28 cores	27	33	43	62	110	200
L-BFGS-B	Intel Core Haswell, no TSX @2.29 GHz-28 cores	2.9	5.1	9.3	18	35	70
m2DLBFGS	Intel Core Haswell, no TSX @2.29 GHz-28 cores	3.7	5.4	9.8	20	36	68
CMA	Intel Core Haswell, no TSX @2.29 GHz-28 cores	17	20	30	49	95	200

Table 1: CPU timing per function evaluation

6 OBSERVATIONS AND CONCLUSION

While the performance of LMCMA and RmES does not change between the original and rotated Ellipsoid function, as well as between the original and rotated Rosenbrock function, this is not the case for VkD-CMA. In particular, in dimension 320 the runtime is larger by at least a factor of 10 for the Ellipsoid function when rotations are applied, as presented in Figure 5. This function is an example that perfectly illustrates the tradeoff of a restricted covariance matrix model that exploits separability and of maintaining rotational invariance: with no rotation, sepCMA is the fastest method, more than 10 times faster than CMA for the most difficult targets. Then VD with one principal search direction, V2D and VkD follow. When rotations are applied, the opposite effect appears, with sepCMA not reaching any target, and VD and V2D showing the worst performance right after sepCMA. The property of invariance under affine transformations of the search space that CMA-ES possesses does not hold for the restricted complexity model methods.

Looking at the aggregated ECDFs of all functions in Figures 2 and 3 in order to compare the number of evolution paths of RmES, the picture is diverse in dimension 80 but more clear in dimension 320, where R2ES dominates R10ES for all budgets. The latter can be of advantage though for certain functions. For the

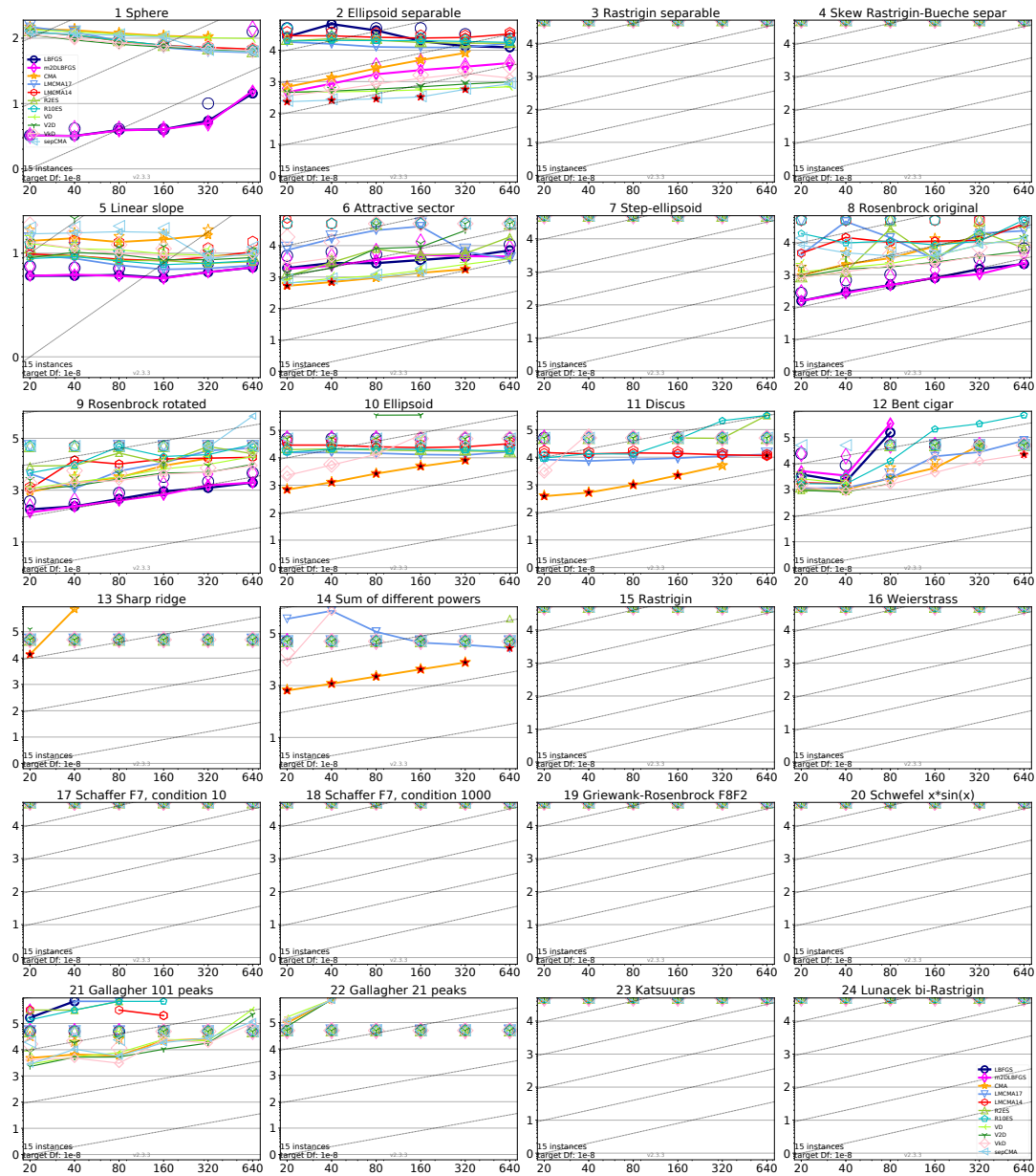


Figure 1: Average running time (aRT in number of f -evaluations as \log_{10} value), divided by dimension for target function value 10^{-8} versus dimension. Slanted grid lines indicate quadratic scaling with the dimension. Different symbols correspond to different algorithms given in the legend of f_1 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: \circ : CMA, \diamond : LBFSGS, \star : LMCMA14, ∇ : LMCMA17, \circ : R10ES, \triangle : R2ES, \square : V2D, \triangleright : VD, ∇ : VxD, \diamond : m2DLBFGS, \triangleleft : sepCMA.

convex quadratic Discus function, in dimensions smaller or equal to 160, R10ES is preferable and in dimension 320 R2ES overtakes R10ES, while the picture is opposite for the Bent Cigar function where R2ES dominates R10ES only in dimension 20. This fact suggests that the parameter strongly relates to the number of short axes and a larger value provides more robustness for these functions. As a result,

R10ES is clearly superior to R2ES on the group of ill-conditioned functions in dimension 80 and the performance difference becomes less significant in dimension 320.

Another observation is the small success rate of LMCMA and RmES for the Gallagher function f_{21} in 320D, illustrated in Figure

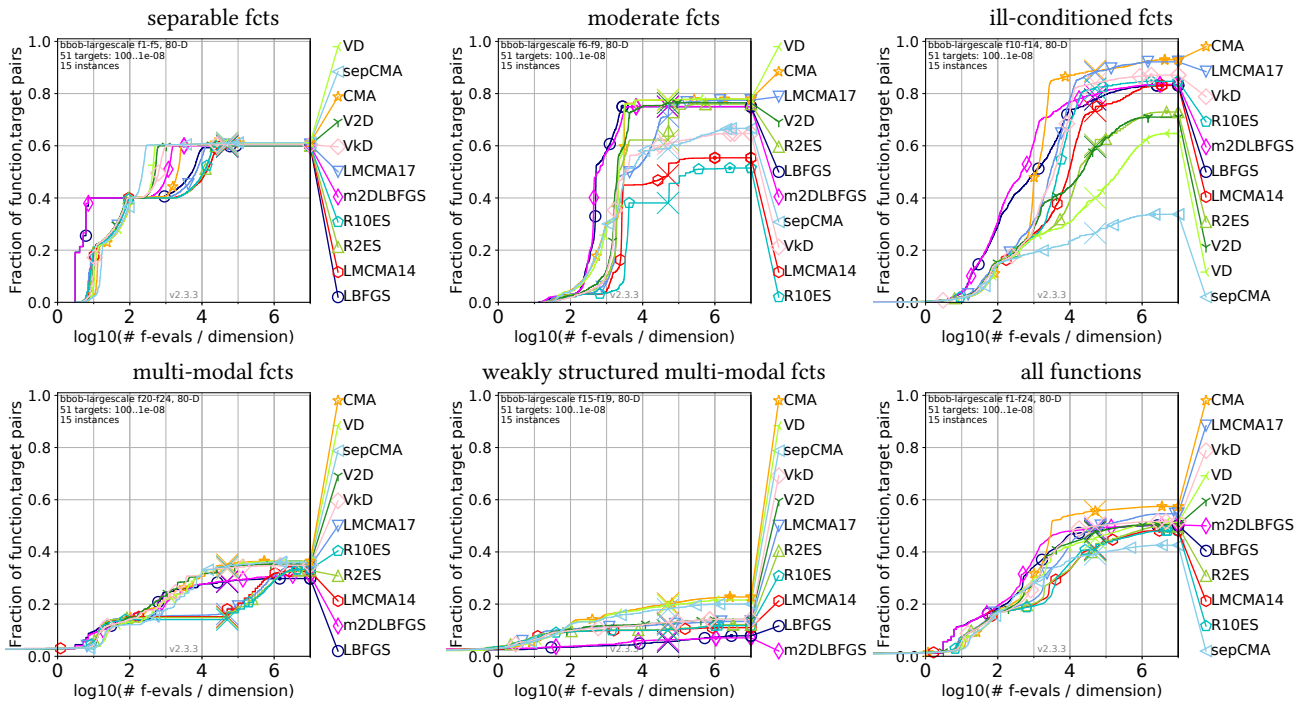


Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 51 targets with target precision in $10^{[-8..2]}$ for all functions and subgroups in 80-D.

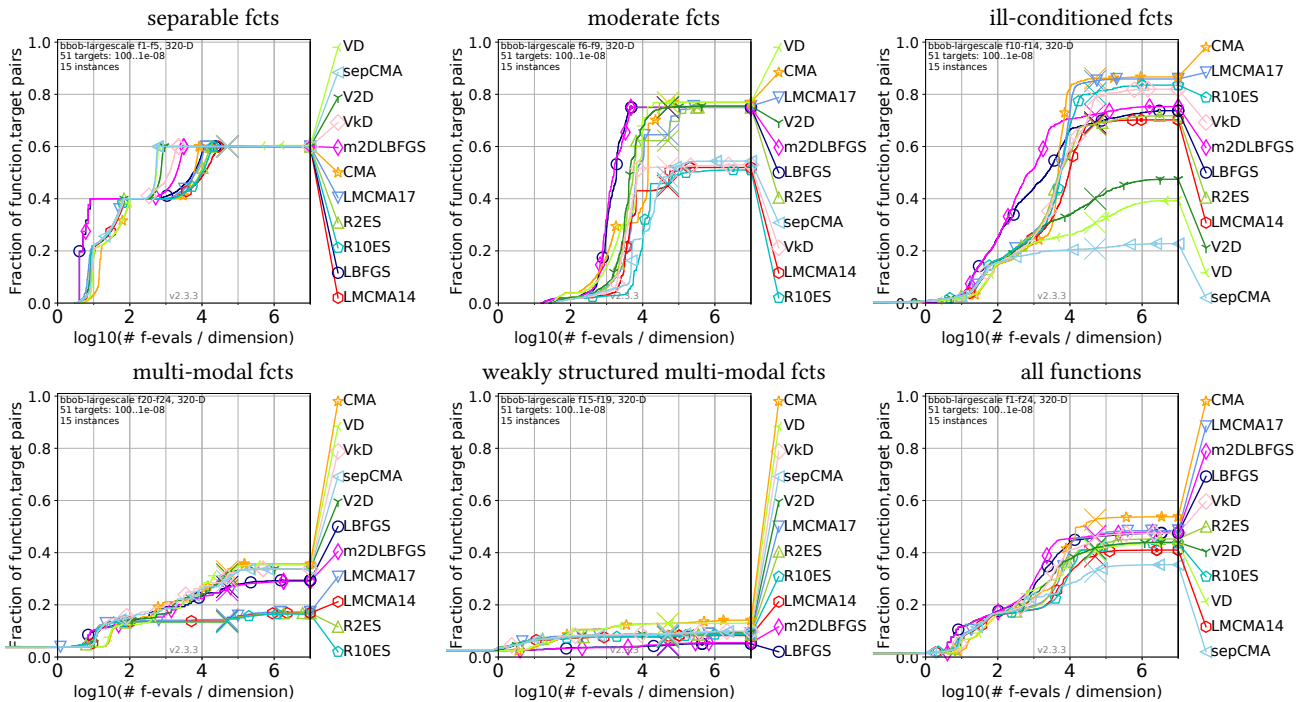


Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 51 targets with target precision in $10^{[-8..2]}$ for all functions and subgroups in 320-D.

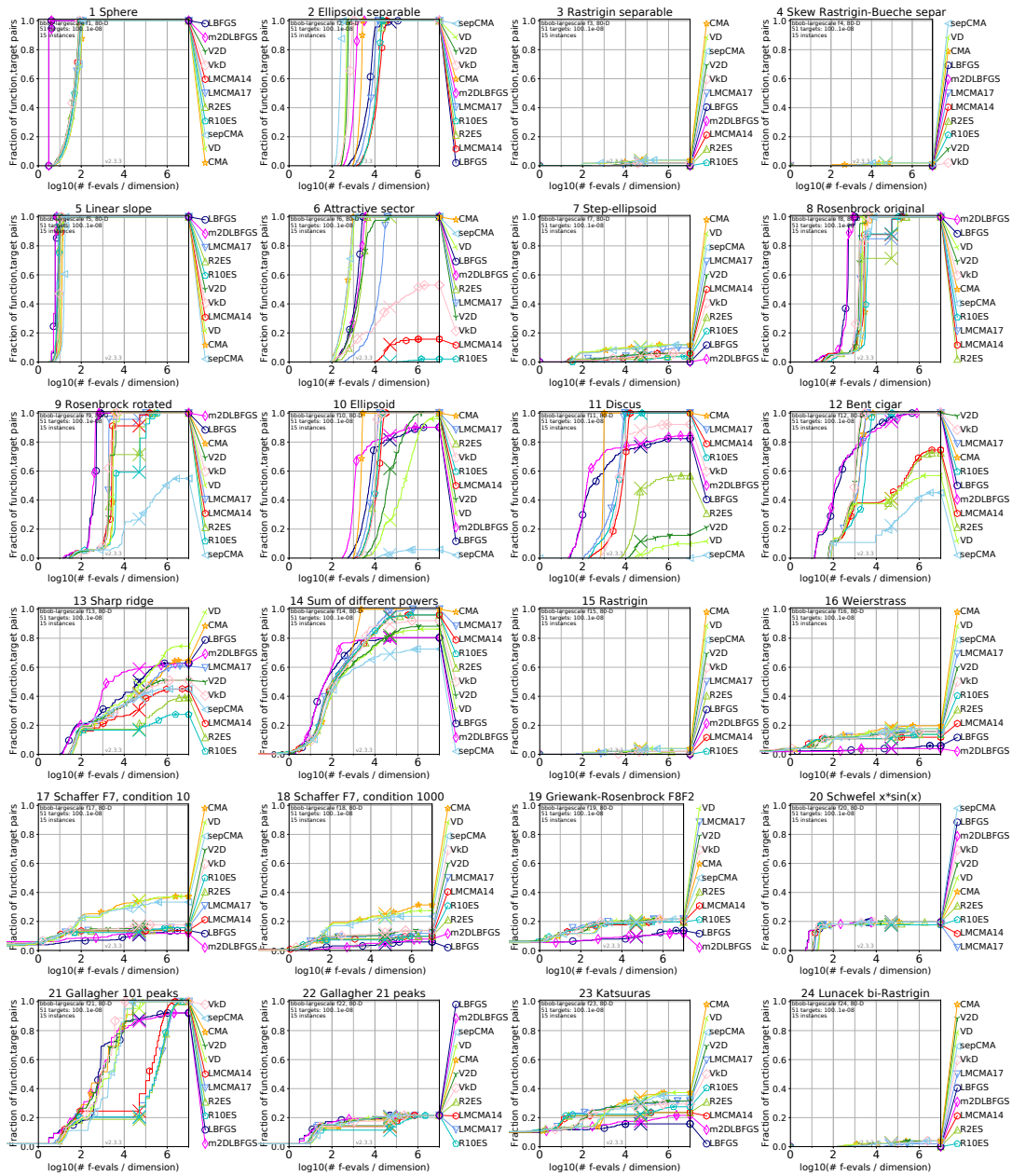


Figure 4: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of objective function evaluations, divided by dimension (FEvals/DIM) for the 51 targets $10^{[-8..2]}$ in dimension 80.

5. This is due to poor termination conditions of the specific implementations that employ only step size values compared to the other benchmarked solvers, for which the high success rate is attributed to the restart policy.

In the case of L-BFGS-B, increasing the maximum number of corrections is clearly of advantage, that affects mostly the group of ill-conditioned functions. Considering the example of the separable Ellipsoid function in dimension 320 depicted in Figure 5, the runtime is smaller by a factor of 2 for the easiest targets and this factor

increases up to 6 for the most difficult targets. This configuration can imply a slight defect, e.g. for the Bent Cigar function, but in overview it dominates the default setting as can be seen from the aggregated ECDFs of Figures 2 and 3, where for all budgets the success rate is superior.

Overall, in all dimensions the best configuration of L-BFGS-B has superior success rate than CMA or the best CMA variant for a restricted budget range. As the budget increases, CMA as well as the best CMA variant overtake L-BFGS-B. That is, for dimensions

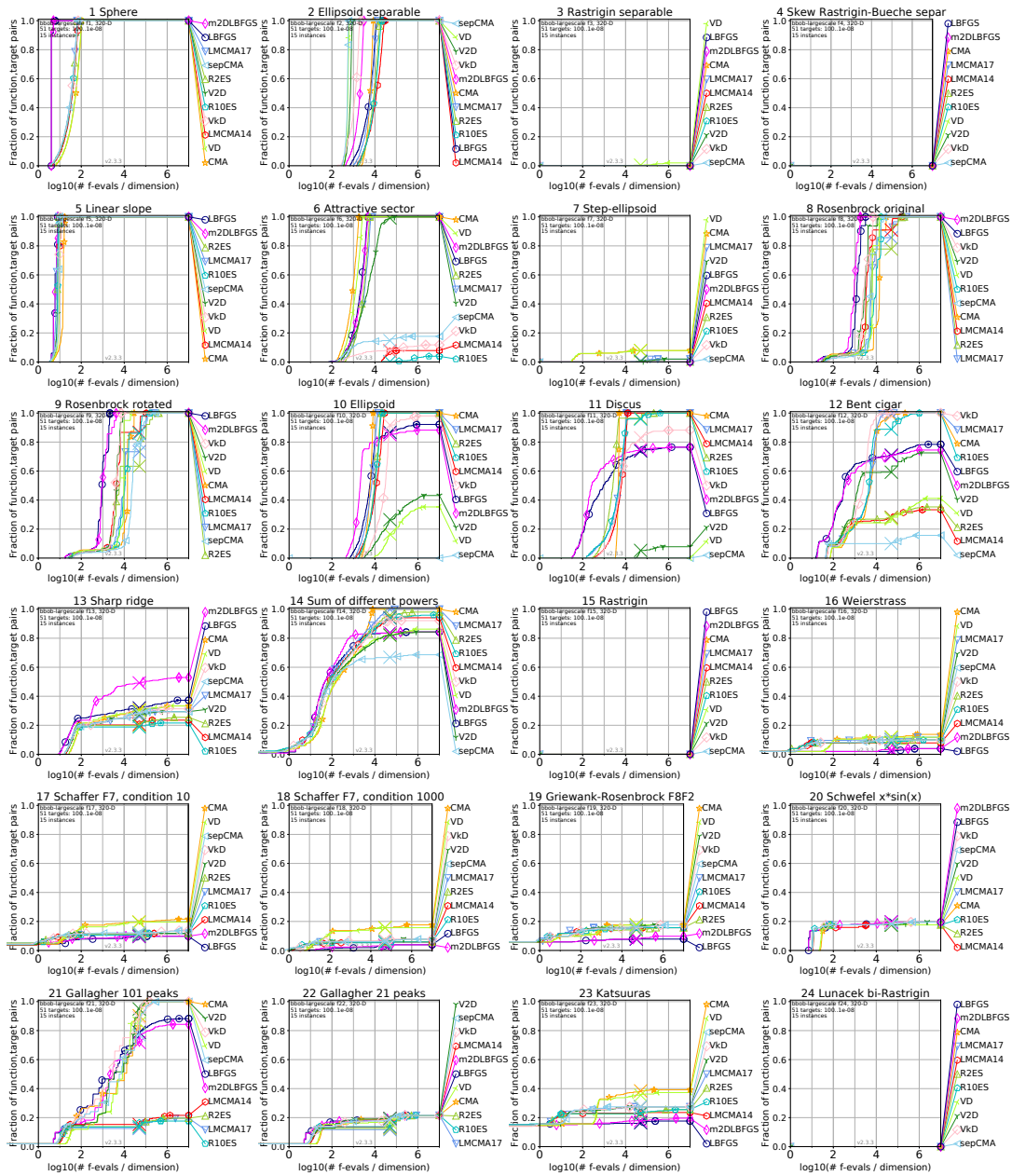


Figure 5: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of objective function evaluations, divided by dimension (FEvals/DIM) for the 51 targets $10^{-8..2}$ in dimension 320.

smaller or equal to 40, Vd-CMA shows the highest success rate right after CMA when a sufficient number of function evaluations is used, and for larger dimensions LMCMA17 is better than Vd-CMA.

7 ACKNOWLEDGEMENTS

The PhD thesis of Konstantinos Varelak is funded by the French MoD DGA/MRIS and Thales Land & Air Systems.

REFERENCES

- [1] Youhei Akimoto, Anne Auger, and Nikolaus Hansen. 2014. Comparison-based natural gradient optimization in high dimension. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, 373–380.
- [2] Youhei Akimoto and Nikolaus Hansen. 2016. Online model selection for restricted covariance matrix adaptation. In *International Conference on Parallel Problem Solving from Nature*. Springer, 3–13.
- [3] Youhei Akimoto and Nikolaus Hansen. 2016. Projection-Based Restricted Covariance Matrix Adaptation for High Dimension. In *Genetic and Evolutionary Computation Conference (GECCO 2016)*. Denver, United States, 197–204.
- [4] Ouassim Ait Elhara, Konstantinos Varelak, Duc Hung Nguyen, Tea Tusar, Dimo Brockhoff, Nikolaus Hansen, and Anne Auger. 2019. COCO: The Large Scale

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ	Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f1																	
LBFG	485(0)	485(0)	485(0)	485(0)	485(0)	485(0)	614(80)	15/15	LBFG	∞	∞	∞	∞	∞	∞	∞	0/15
m2DL	485(0)	485(0)	485(0)	485(0)	485(0)	485(0)	614(80)	15/15	m2DL	∞	∞	∞	∞	∞	∞	∞	0/15
CMA	2697(111)	4364(146)	5961(102)	7631(108)	9224(130)	1.2e4(192)	1.6e4(182)	15/15	CMA	∞	∞	∞	∞	∞	∞	∞	0/15
17lm	1865(49)	2950(85)	4006(147)	5081(155)	6113(150)	8256(158)	1.0e4(132)	15/15	17lm	∞	∞	∞	∞	∞	∞	∞	0/15
14lm	1946(109)	3071(58)	4220(170)	5384(155)	6513(207)	8877(185)	1.1e4(156)	15/15	14lm	∞	∞	∞	∞	∞	∞	∞	0/15
R2ES	1774(84)	2960(164)	4082(134)	5199(161)	6318(209)	8732(161)	1.1e4(243)	15/15	R2ES	∞	∞	∞	∞	∞	∞	∞	0/15
R10E	1886(150)	3053(166)	4294(154)	5391(143)	6627(160)	8961(249)	1.1e4(304)	15/15	R10E	∞	∞	∞	∞	∞	∞	∞	0/15
VD	2676(136)	4265(284)	5860(158)	7504(168)	9114(188)	1.2e4(232)	1.5e4(303)	15/15	VD	∞	∞	∞	∞	∞	∞	∞	0/15
V2D	1728(156)	2797(237)	3895(215)	5018(233)	6099(359)	8290(272)	1.1e4(388)	15/15	V2D	∞	∞	∞	∞	∞	∞	∞	0/15
VkD	1739(95)	2807(122)	3891(212)	5041(264)	6172(369)	8508(296)	1.1e4(517)	15/15	VkD	∞	∞	∞	∞	∞	∞	∞	0/15
sepC	2577(50)	4157(110)	5703(116)	7280(160)	8820(125)	1.2e4(146)	1.5e4(238)	15/15	sepC	∞	∞	∞	∞	∞	∞	∞	0/15
f8																	
LBFG	2.5e5(2e4)	4.0e5(2e4)	5.8e5(7e4)	7.4e5(1e5)	9.2e5(1e5)	1.3e6(1e5)	1.6e6(4e5)	13/15	LBFG	7.8e4(2e4)	9.6e4(5e4)	1.2e5(2e5)	1.2e5(2e5)	1.2e5(6e4)	1.3e5(2e5)	1.3e5(5e4)	15/15
m2DL	1.2e5(4830)	1.5e5(7004)	1.9e5(7245)	2.1e5(8936)	2.5e5(1e4)	2.9e5(1e4)	3.3e5(8251)	15/15	m2DL	8.5e4(2e4)	1.1e5(2e4)	1.2e5(5e4)	1.2e5(6e4)	1.3e5(3e4)	1.3e5(7e4)	1.3e5(6e4)	15/15
CMA	3.7e5(2e4)	4.4e5(2e4)	5.1e5(1e4)	5.6e5(2e4)	6.1e5(2e4)	6.8e5(1e4)	7.5e5(3e4)	15/15	CMA	7.3e5(2e4)	9.4e5(2e5)	1.2e6(5e5)	1.2e6(5e5)	1.2e6(5e5)	1.2e6(3e5)	1.2e6(6e5)	15/15
17lm	3.5e5(4e4)	5.3e5(5e4)	7.1e5(5e4)	8.8e5(1e5)	1.1e6(8e4)	1.4e6(1e5)	1.8e6(3e5)	15/15	17lm	3.4e5(3e4)	4.3e5(1e5)	4.4e5(1e5)	4.5e5(2e5)	4.6e5(1e5)	4.7e5(1e5)	4.7e5(1e5)	15/15
14lm	5.3e5(6e4)	8.6e5(1e5)	1.2e6(2e5)	1.5e6(4e5)	1.9e6(3e5)	2.7e6(4e5)	3.5e6(4e5)	15/15	14lm	3.6e5(9e4)	4.9e5(2e5)	1.7e6(4e6)	1.8e6(1e5)	1.8e6(2e5)	1.8e6(4e6)	1.8e6(2e5)	13/15
R2ES	5.9e5(6e4)	8.6e5(7e4)	1.1e6(9e4)	1.3e6(7e4)	1.6e6(9e4)	2.1e6(1e5)	2.6e6(1e5)	15/15	R2ES	3.5e5(1e5)	4.6e5(8e4)	4.7e5(1e5)	4.8e5(1e5)	4.8e5(1e5)	4.8e5(2e5)	4.9e5(2e5)	15/15
R10E	5.5e5(4e4)	8.3e5(5e4)	1.1e6(6e4)	1.4e6(1e5)	1.7e6(8e4)	2.3e6(2e5)	2.9e6(3e5)	15/15	R10E	5.6e5(3e5)	7.0e5(3e5)	1.3e6(6e6)	1.3e6(4e5)	1.3e6(2e6)	1.3e6(2e6)	1.3e6(2e6)	14/15
VD	6.1e4(4112)	6.8e4(5968)	7.2e4(5928)	7.5e4(3457)	7.8e4(3953)	8.2e4(4758)	8.5e4(4732)	15/15	VD	4.2e5(9e62)	5.5e5(1e4)	6.0e5(9102)	6.1e5(8947)	6.2e5(3e5)	6.2e5(3e5)	6.3e5(1e4)	15/15
V2D	6.5e4(3328)	7.6e4(1213)	8.5e4(4990)	9.1e4(5302)	9.5e4(4733)	1.0e5(4184)	1.1e5(2758)	15/15	V2D	2.5e5(5e4)	3.1e5(9e4)	3.7e5(2e5)	3.7e5(8e4)	3.8e5(8e4)	3.9e5(7e4)	3.9e5(2e5)	15/15
VkD	6.9e4(2e4)	9.9e4(3e4)	1.2e5(1e4)	1.4e5(2e4)	1.5e5(2e4)	1.8e5(3e4)	1.9e5(5e4)	15/15	VkD	2.5e5(5e4)	3.1e5(9e4)	3.3e5(1e5)	3.4e5(1e5)	3.4e5(1e5)	3.5e5(1e5)	3.5e5(8e4)	15/15
sepC	3.3e4(2902)*4	3.7e4(2168)*4	3.9e4(2958)*4	4.2e4(2338)*4	4.4e4(3288)*4	4.7e4(3049)*4	5.0e4(2160)*4	15/15	sepC	3.1e5(1e4)	4.1e5(2e4)	5.3e5(3e5)	5.4e5(3e5)	5.5e5(1e5)	5.8e5(3e5)	6.1e5(1e5)	15/15
f3																	
LBFG	∞	∞	∞	∞	∞	∞	∞	0/15	LBFG	7.2e4(3e4)	8.5e4(3e4)	1.4e5(2e5)	1.4e5(4e4)	1.4e5(2e5)	1.4e5(2e5)	1.4e5(2e4)	15/15
m2DL	∞	∞	∞	∞	∞	∞	∞	0/15	m2DL	7.9e4(3e4)	9.4e4(3e4)	1.0e5(1e5)	1.1e5(5e4)	1.1e5(5e4)	1.1e5(1e5)	1.1e5(6e4)	15/15
CMA	∞	∞	∞	∞	∞	∞	∞	0/15	CMA	7.4e5(1e4)	1.0e6(3e4)	1.4e6(5e5)	1.4e6(3e5)	1.4e6(5e5)	1.4e6(3e5)	1.4e6(5e5)	15/15
17lm	∞	∞	∞	∞	∞	∞	∞	0/15	17lm	3.9e5(2e4)	5.3e5(3e4)	1.8e6(2e6)	1.8e6(1e5)	1.8e6(6e6)	1.8e6(6e6)	1.8e6(4e6)	13/15
14lm	∞	∞	∞	∞	∞	∞	∞	0/15	14lm	4.1e5(8e4)	5.4e5(1e5)	2.5e6(8e6)	2.5e6(4e6)	2.5e6(4e6)	2.5e6(6e6)	2.5e6(6e6)	12/15
R2ES	∞	∞	∞	∞	∞	∞	∞	0/15	R2ES	3.6e5(5e4)	4.8e5(7940)	1.7e6(4e6)	1.7e6(4e6)	1.8e6(2e6)	1.8e6(4e6)	1.8e6(4e6)	13/15
R10E	∞	∞	∞	∞	∞	∞	∞	0/15	R10E	8.9e5(2e4)	1.1e6(2e5)	3.2e6(6e6)	3.2e6(6e6)	3.2e6(2e6)	3.2e6(6e6)	3.2e6(6e6)	12/15
VD	∞	∞	∞	∞	∞	∞	∞	0/15	VD	5.9e5(3e4)	7.8e5(1e5)	1.0e6(9e5)	1.0e6(9e5)	1.0e6(4e5)	1.0e6(2e5)	1.0e6(4e5)	15/15
V2D	∞	∞	∞	∞	∞	∞	∞	0/15	V2D	3.6e5(5e4)	4.9e5(8e4)	7.1e5(5e5)	7.2e5(5e5)	7.2e5(3e5)	7.3e5(3e5)	7.3e5(3e5)	15/15
VkD	∞	∞	∞	∞	∞	∞	∞	0/15	VkD	3.6e5(9e4)	4.6e5(8e4)	6.4e5(3e5)	6.4e5(3e5)	6.5e5(2e5)	6.6e5(3e5)	6.6e5(4e5)	15/15
sepC	∞	∞	∞	∞	∞	∞	∞	0/15	sepC	2.0e6(9e4)	1.1e8(1e8)	1.1e8(1e8)	1.1e8(2e8)	1.1e8(1e8)	∞	∞	0/15
f10																	
LBFG	∞	∞	∞	∞	∞	∞	∞	0/15	LBFG	2.5e5(3e4)	3.9e5(4e4)	5.6e5(6e4)	7.2e5(1e5)	9.1e5(9e4)	1.3e6(2e5)	∞	0/15
m2DL	∞	∞	∞	∞	∞	∞	∞	0/15	m2DL	1.2e5(5072)*4	1.6e5(7084)*4	1.9e5(7849)*4	2.1e5(6198)*4	2.4e5(5554)*4	3.9e5(1e5)*4	1.1e8(1e8)	0/15
CMA	∞	∞	∞	∞	∞	∞	∞	0/15	CMA	3.7e5(2e4)	4.4e5(4e4)	5.0e5(1e4)	5.6e5(9718)	6.0e5(2e4)	6.8e5(2e4)	7.5e5(2e4)*2	15/15
17lm	∞	∞	∞	∞	∞	∞	∞	0/15	17lm	3.7e5(4e4)	5.6e5(3e4)	7.6e5(8e4)	9.5e5(1e5)	1.1e6(1e5)	1.5e6(2e5)	1.9e6(2e5)	15/15
14lm	∞	∞	∞	∞	∞	∞	∞	0/15	14lm	5.1e5(7e4)	8.6e5(1e5)	1.2e6(2e5)	1.5e6(2e5)	1.9e6(3e5)	3.4e6(5e5)	3.4e6(5e5)	15/15
R2ES	∞	∞	∞	∞	∞	∞	∞	0/15	R2ES	5.8e5(7e4)	8.4e5(7e4)	1.1e6(1e5)	1.4e6(1e5)	1.6e6(2e5)	2.1e6(3e5)	2.7e6(2e5)	15/15
R10E	∞	∞	∞	∞	∞	∞	∞	0/15	R10E	5.5e5(3e4)	8.1e5(9e4)	1.1e6(2e5)	1.4e6(3e5)	1.7e6(2e5)	2.3e6(3e5)	3.0e6(3e5)	15/15
VD	∞	∞	∞	∞	∞	∞	∞	0/15	VD	3.3e6(2e6)	5.5e6(7e6)	1.1e7(8e6)	2.8e7(5e7)	1.2e8(2e8)	∞	∞	0/15
V2D	∞	∞	∞	∞	∞	∞	∞	0/15	V2D	1.6e6(7e5)	3.1e6(2e6)	4.8e6(3e6)	7.4e6(3e6)	1.7e7(1e7)	5.6e7(7e7)	5.7e7(6e7)	2/15
VkD	∞	∞	∞	∞	∞	∞	∞	0/15	VkD	1.0e6(4e5)	1.6e6(3e5)	2.1e6(4e5)	2.5e6(5e5)	2.9e6(5e5)	3.4e6(6e5)	3.7e6(3e5)	9/15
sepC	∞	∞	∞	∞	∞	∞	∞	0/15	sepC	∞	∞	∞	∞	∞	∞	∞	0/15
f11																	
LBFG	764(121)	861(80)	925(80)	925(80)	925(0)	925(80)	925(121)	15/15	LBFG	1.1e4(2777)	1.7e4(3582)	2.4e4(5152)	3.5e4(2e4)	1.1e5(9e4)	2.0e6(2e6)	∞	0/15
m2DL	764(121)	861(121)	904(121)	914(121)	914(80)	914(161)	914(80)	15/15	m2DL	1.1e4(3381)	1.7e4(2858)	2.3e4(5716)	2.7e4(6561)	4.1e4(3e4)*2	4.1e5(7e5)	∞	0/15
CMA	1648(148)	2019(216)	2157(328)	2171(234)	2172(165)	2172(164)	2172(164)	15/15	CMA	2.7e5(5103)	2.9e5(805)	3.0e5(6118)	3.1e5(3660)	3.2e5(4136)	3.4e5(3404)	3.5e5(2715)*4	0/15
17lm	977(146)	1071(116)	1108(178)	1113(192)	1116(126)	1116(164)	1116(160)	15/15	17lm	7.7e4(7665)	1.9e5(1e4)	3.5e5(3e4)	5.1e5(2e4)	6.8e5(3e4)	1.0e6(3e4)	1.3e6(4e4)	15/15
14lm	1181(150)	1310(192)	1348(214)	1353(138)	1354(200)	1354(188)	1354(237)	15/15	14lm	2.5e5(6e4)	4.6e5(5e4)	6.6e5(3e4)	9.0e5(5e4)	1.1e6(9e4)	1.5e6(1e5)	2.0e6(1e5)	15/15
R2ES	1146(70)	1310(125)	1349(88)	1352(119)	1358(79)	1358(64)	1358(95)	15/15	R2ES	6.8e5(1e5)	1.1e6(1e5)	1.4e6(2e5)	1.8e6(1e5)	2.2e6(2e5)	2.9e6(2e5)	3.6e6(3e5)	10/15
R10E	1090(122)	1196(139)	1236(138)	1240(175)	1240(128)	1240(128)	1240(144)	15/15	R10E	1.4e5(4e4)	3.4e5(7e4)	5.4e5(9e4)	7.4e5(1e5)	9.4e5(1e5)	1.3e6(2e5)	1.7e6(2e5)	9/15
VD	1323(132)	1515(129)	1556(138)	1574(146)	1577(154)	1577(142)	1577(162)	15/15	VD	∞	∞	∞	∞	∞	∞	∞	0/15
V2D	1178(45)	1332(203)	1373(122)	1383(245)	1383(286)	1383(346)	1383(214)	15/15	V2D	3.7e7(4e7)	∞	∞	∞	∞	∞	∞	0/15
VkD	1217(167)	1420(132)	1499(119)	1521(127)	1521(129)	1521(172)	1521(125)	15/15	VkD	2.7e5(7e4)	4.9e5(2e5)	7.0e5(2e5)	8.9e5(4e5)	1.1e6(5e5)	1.8e6(3e5)	1.1e8(2e8)	0/15
sepC	1810(101)	2321(164)	2500(141)	2525(98)	2531(200)	2531(204)	2531(91)	15/15	sepC	∞	∞	∞					

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ	Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
f13																	
LBFG	3920(201)	4.1e4(5e4)	2.5e6(3e6)	5.7e7(4e7)	∞	∞	∞	0/15	LBFG	∞	∞	∞	∞	∞	∞	∞	0/15
m2DL	3952(282)	4.0e4(8895)	1.9e5(4e5) *	3.3e5(3e5) * ²	4.6e6(1e7) * ⁴	∞	∞	0/15	m2DL	∞	∞	∞	∞	∞	∞	∞	0/15
CMA	9761(362)	6.0e4(7e4)	6.2e6(4e6)	∞	∞	∞	∞	0/15	CMA	4096(2157)	∞	∞	∞	∞	∞	∞	0/15
17lm	6418(251)	1.8e5(2e5)	9.7e6(1e7)	1.1e8(1e8)	∞	∞	∞	0/15	17lm	1251(269)	∞	∞	∞	∞	∞	∞	0/15
14lm	7388(456)	7.3e6(8e6)	3.5e7(4e7)	∞	∞	∞	∞	0/15	14lm	1426(239)	∞	∞	∞	∞	∞	∞	0/15
R2ES	6910(527)	3.2e7(2e7)	∞	∞	∞	∞	∞	0/15	R2ES	1330(161)	∞	∞	∞	∞	∞	∞	0/15
R10E	7512(804)	1.1e8(7e7)	∞	∞	∞	∞	∞	0/15	R10E	1359(206)	∞	∞	∞	∞	∞	∞	0/15
VD	9448(360)	1.1e5(1e5)	4.0e6(5e6)	∞	∞	∞	∞	0/15	VD	3558(1365)	∞	∞	∞	∞	∞	∞	0/15
V2D	7253(651)	2.4e5(4e5)	7.8e6(8e6)	∞	∞	∞	∞	0/15	V2D	4989(2233)	∞	∞	∞	∞	∞	∞	0/15
VkD	7454(709)	1.9e5(4e5)	8.8e6(1e7)	∞	∞	∞	∞	0/15	VkD	3987(2615)	∞	∞	∞	∞	∞	∞	0/15
sepC	8788(410)	8.7e4(1e5)	6.5e6(5e6)	∞	∞	∞	∞	0/15	sepC	4198(2338)	∞	∞	∞	∞	∞	∞	0/15
f14																	
LBFG	914(161)	1913(282)	2793(523)	4231(564)	7730(402)	7.9e4(7245)	∞	0/15	LBFG	1226(80)	∞	∞	∞	∞	∞	∞	0/15
m2DL	957(161)	2063(322)	2771(402)	4274(322)	8063(483)	3.5e4(604) * ⁴	∞	0/15	m2DL	1226(80)	∞	∞	∞	∞	∞	∞	0/15
CMA	2407(215)	5163(342)	7698(171)	1.3e4(538)	3.0e4(1115)	1.6e5(7500)	4.7e5(1e4) * ⁴	15/15	CMA	4211(198)	∞	∞	∞	∞	∞	∞	0/15
17lm	1302(126)	3100(267)	4439(250)	6595(383)	1.3e4(805)	1.1e5(4497)	1.6e6(8e4)	14/15	17lm	2085(154)	∞	∞	∞	∞	∞	∞	0/15
14lm	1418(119)	3507(164)	5133(186)	7982(198)	1.5e4(856)	1.4e5(6753)	3.2e6(2e5)	0/15	14lm	2070(109)	∞	∞	∞	∞	∞	∞	0/15
R2ES	1338(133)	3267(150)	4819(135)	7442(429)	1.6e4(519)	1.6e5(1e4)	2.5e6(1e5)	0/15	R2ES	1850(82)	∞	∞	∞	∞	∞	∞	0/15
R10E	1280(121)	3440(291)	5069(323)	8172(320)	1.7e4(1045)	1.5e5(1e4)	3.5e6(4e5)	0/15	R10E	2007(78)	∞	∞	∞	∞	∞	∞	0/15
VD	2238(395)	4940(331)	7537(321)	1.3e4(437)	2.6e4(1740)	3.6e5(1e5)	∞	0/15	VD	3793(359)	∞	∞	∞	∞	∞	∞	0/15
V2D	1357(220)	3491(406)	5131(264)	8160(322)	1.8e4(904)	3.5e5(1e5)	∞	0/15	V2D	1876(107)	∞	∞	∞	∞	∞	∞	0/15
VkD	1457(138)	3607(428)	5261(407)	8789(172)	1.9e4(1849)	2.0e5(5e4)	3.6e6(4e6)	0/15	VkD	1988(200)	∞	∞	∞	∞	∞	∞	0/15
sepC	2264(268)	4854(375)	7247(464)	1.2e4(468)	2.4e4(2136)	1.5e6(5e5)	∞	0/15	sepC	3858(228)	∞	∞	∞	∞	∞	∞	0/15
f15																	
LBFG	∞	∞	∞	∞	∞	∞	∞	0/15	LBFG	3877(6602)	1.1e6(2e6)	2.2e6(2e6)	2.2e6(4e6)	2.2e6(3e6)	2.2e6(4e6)	1.2e8(1e8)	0/15
m2DL	∞	∞	∞	∞	∞	∞	∞	0/15	m2DL	4038(7367)	4.9e5(2e6)	9.8e5(2e5)	9.8e5(2e6)	9.9e5(2e5)	9.9e5(1e6)	∞	0/15
CMA	∞	∞	∞	∞	∞	∞	∞	0/15	CMA	1.2e4(1e4)	1.2e6(4e6)	3.5e6(5e6)	3.5e6(6e6)	3.5e6(5e6)	3.5e6(7e6)	12/15	
17lm	∞	∞	∞	∞	∞	∞	∞	0/15	17lm	5.3e6(8e6)	∞	∞	∞	∞	∞	0/15	
14lm	∞	∞	∞	∞	∞	∞	∞	0/15	14lm	2.0e6(4e6)	2.2e7(2e7)	3.2e7(4e7)	3.2e7(4e7)	3.2e7(3e7)	3.2e7(4e7)	3/15	
R2ES	∞	∞	∞	∞	∞	∞	∞	0/15	R2ES	1.2e6(1326)	1.1e8(7e7)	∞	∞	∞	∞	0/15	
R10E	∞	∞	∞	∞	∞	∞	∞	0/15	R10E	1.2e6(299)	5.2e7(6e7)	1.1e8(2e8)	1.1e8(7e7)	1.1e8(1e8)	1.1e8(1e8)	1/15	
VD	∞	∞	∞	∞	∞	∞	∞	0/15	VD	8017(2e4)	1.6e6(6e6)	3.9e6(8e6)	3.9e6(8e6)	4.0e6(6e6)	4.0e6(1e7)	11/15	
V2D	∞	∞	∞	∞	∞	∞	∞	0/15	V2D	1.3e4(3e4)	4.4e5(5e5)	1.7e6(3e6)	1.7e6(4e6)	1.7e6(4e6)	1.7e6(2e6)	14/15	
VkD	∞	∞	∞	∞	∞	∞	∞	0/15	VkD	1.3e4(1e4)	1.0e6(1e5)	3.9e6(6e6)	3.9e6(4e6)	3.9e6(8e6)	3.9e6(4e6)	11/15	
sepC	∞	∞	∞	∞	∞	∞	∞	0/15	sepC	8201(3e4)	7.6e5(7e5)	3.0e6(2e6)	3.0e6(6e6)	3.0e6(5e6)	3.0e6(3e6)	13/15	
f16																	
LBFG	∞	∞	∞	∞	∞	∞	∞	0/15	LBFG	2.1e4(1e5)	5.9e6(1e7)	∞	∞	∞	∞	∞	0/15
m2DL	∞	∞	∞	∞	∞	∞	∞	0/15	m2DL	3.2e4(4026)	5.7e6(7e6)	∞	∞	∞	∞	∞	0/15
CMA	4.6e4(2e5)	∞	∞	∞	∞	∞	∞	0/15	CMA	8.3e4(2e5)	7.7e6(6e6)	∞	∞	∞	∞	∞	0/15
17lm	9.7e4(9e4)	∞	∞	∞	∞	∞	∞	0/15	17lm	4.0e6(8e6)	5.2e7(7e7)	∞	∞	∞	∞	0/15	
14lm	1.1e8(2e8)	∞	∞	∞	∞	∞	∞	0/15	14lm	5.3e6(8e6)	2.2e7(1e7)	∞	∞	∞	∞	0/15	
R2ES	2.2e7(2e7)	∞	∞	∞	∞	∞	∞	0/15	R2ES	5.3e6(8e6)	3.2e7(3e7)	∞	∞	∞	∞	0/15	
R10E	5.2e7(6e7)	∞	∞	∞	∞	∞	∞	0/15	R10E	2.0e6(2e6)	1.1e8(8e7)	∞	∞	∞	∞	0/15	
VD	1.8e4(3650)	∞	∞	∞	∞	∞	∞	0/15	VD	6.0e4(8e4)	7.6e6(6e6)	∞	∞	∞	∞	0/15	
V2D	1.0e6(1e6)	∞	∞	∞	∞	∞	∞	0/15	V2D	1.5e5(5e5)	6.7e6(2e7)	∞	∞	∞	∞	0/15	
VkD	1.9e6(2e6)	∞	∞	∞	∞	∞	∞	0/15	VkD	2.1e4(6e4)	7.0e6(1e7)	∞	∞	∞	∞	0/15	
sepC	2.9e5(2e5)	∞	∞	∞	∞	∞	∞	0/15	sepC	3.6e4(6e4)	5.4e6(6e6)	∞	∞	∞	∞	0/15	
f17																	
LBFG	1.1e8(1e8)	∞	∞	∞	∞	∞	∞	0/15	LBFG	1(0)	∞	∞	∞	* ⁴	* ⁴	∞	0/15
m2DL	2.0e7(2e7)	∞	∞	∞	∞	∞	∞	0/15	m2DL	1(0)	∞	∞	∞	∞	∞	∞	0/15
CMA	3126(502)	8.2e4(8e4)	∞	∞	∞	∞	∞	0/15	CMA	57(0)	1.0e5(8000)	7.2e5(5e5)	∞	∞	∞	∞	0/15
17lm	1509(168)	∞	∞	∞	∞	∞	∞	0/15	17lm	1(0)	1622(211)	∞	∞	∞	∞	0/15	
14lm	1.2e6(7778)	∞	∞	∞	∞	∞	∞	0/15	14lm	1(0)	5172(1e4)	∞	∞	∞	∞	0/15	
R2ES	1.2e6(2e6)	∞	∞	∞	∞	∞	∞	0/15	R2ES	1(0)	2.9e6(8e6)	∞	∞	∞	∞	0/15	
R10E	3759(4721)	∞	∞	∞	∞	∞	∞	0/15	R10E	1(0)	2.0e6(2e6)	∞	∞	∞	∞	0/15	
VD	2810(396)	1.5e5(2e5)	∞	∞	∞	∞	∞	0/15	VD	22(157)	1.1e5(4e4)	2.7e6(3e6)	∞	∞	∞	0/15	
V2D	1922(733)	∞	∞	∞	∞	∞	∞	0/15	V2D	1(0)	8922(5388)	∞	∞	∞	∞	0/15	
VkD	2524(1932)	∞	∞	∞	∞	∞	∞	0/15	VkD	6.3(0)	7862(3415)	∞					

- [16] Raymond Ros and Nikolaus Hansen. 2008. A Simple Modification in CMA-ES Achieving Linear Time and Space Complexity. In *Parallel Problem Solving from Nature (PPSN 2008)*. Springer, 296–305.
- [17] Thorsten Suttrop, Nikolaus Hansen, and Christian Igel. 2009. Efficient covariance matrix update for variable metric evolution strategies. *Machine Learning* 75, 2 (2009), 167–197.