Experiments on the CEC 2015 expensive optimization testbed

Vincent Berthier

Experiments on the CEC 2015 expensive optimization testbed

Vincent Berthier
TAO, Inria, Univ. Paris-Sud, UMR CNRS 8623
Bat 660 Claude Shannon Univ. Paris-Sud, 91190 Gif-sur-Yvette, France
Email: {*firstname.lastname*}@*inria.fr*

***Abstract*—We experiment various simple classical algorithms on the expensive optimization testbed from Cec2015. CMA performs best, in particular its DCMA flavor using quasi-random numbers. Nelder-Mead also performs well. Portfolios performed well for the given budget (500 evaluations in dimension 10 and 1500 evaluations in dimension 30), but not the half budget, which is also taken into account in the competition, hence we did not include them in the final version.**

Keywords*—*Expensive, Quasi-Random, Optimisation

I. Introduction

[1] proposed a testbed for expensive optimization. This means that the number of fitness evaluations is limited. In this case the budget is 500 in dimension 10 and 1500 in dimension 30. The testcase is also designed for difficult objective functions, with 2 unimodal and 13 multimodal functions including hybrid and composition functions.

[2] proposed the use of portfolio methods in noise-free optimization. Portfolio are generic tools for combining optimization algorithms, most widely used in combinatorial optimization, but also appearing in noisy continuous optimization[3] and noise-free continuous optimization[2].

We also use quasi-random numbers in mutations, as proposed in [4]. Basically, it makes algorithms slightly better on average, with a different distribution, as detailed later.

The computation time in portfolio algorithms can be simply divided equally among solvers, or not[5]. [6], [7] propose 50% for the best solver, 25% for the second best, and so on. One might also run all solvers during e.g. 25% of total time, and then keep 75% of the budget only for the best performing one. This is the approach we keep for all our portfolio experiments in this paper.

Surrogate models are classical for expensive optimization[8]; population-based methods have also been widely used[9], as well as derivative-free methods as expensive optimization is often due to heavy simulations without gradient[10]. Gaussian processes are also widely used as their internal cost becomes negligible in front of the cost of the objective function[11], [12]. We consider mostly population-based optimization; the comparison with the results of other methods will be outputs of the session, comparing various methods on this same testbed.

II. Algorithms

A. Individual algorithms

We use the following implementation in our comparison (N is the dimension):

- Covariance Matrix Adaptation (CMA)[13], $\lambda = 4 + 3\log(N)$, $\mu = \lambda/2$.
- anisotropic self-adaptive[14], $\lambda = 12$, $\mu = 3$, $\tau = 1/\sqrt{2N}$.
- Self-adaptive (SA) [14], $\lambda = 12$, $\mu = 3$, $\tau = 1/\sqrt{2N}$, $\tau_C = 1 + N(N+1)/2\mu$.
- Self-adaptive (SA) with covariance[15], $\lambda = 12$, $\mu = 3$, $\tau = 1/\sqrt{2N}$, $\tau_C = 1 + N(N+1)/2\mu$.
- Differential evolution, DE[16], population size 30, DE/Curr-to-best/1, $Cr = .5$, $F_1 = F_2 = .8$.
- Covariance matrix self-adaptation CMSA[17], $\lambda = 12$, $\mu = 3$, $\tau = 1/\sqrt{2N}$.
- $(1+1)$-ES[15], step-size multiplied by 1.5 in case of success and one-fifth rule (i.e. division by $1.5^{1/4}$ in case of failure).
- Nelder-Mead[18].
- PSO[19], [20] with population size 30, neighborhoud 10, $\omega = 1/(2\log(2))$, $\phi_p = .5 + \log(2)$, $\phi_g = .5 + \log(2)$, initial velocity 1 and maximum velocity 1.5.

The initialization is uniform in the domain, for each algorithm. Due to space restrictions, algorithms are not detailed here, but given references should abswer any question in that regard.

B. Restart/portfolio

[2] reported excellent results for sophisticated methods, and indeed also good stable results with simple methods. We decided to focus on simple methods. In all our experiments, the portfolio equally divides the computation time among the algorithms during 25% of the budget, and then uses only the solver which provided the best search point during the remaining 75% of the budget. CMA/CMA/CMA for example refer to running 3 instances of CMA during 25% of the budget, and then the best performing one during the remaining 75%.

III. Experiments

In the following sections, we first compare several classical algorithms, before tuning the best one and adding quasi-random numbers.

A. *Preliminary experiments: comparing various algorithms*

Fig. I, II, III present the results for dimension 10 and 30 for the worst, median and average rank over the test cases.

A short discussion is that CMA and NM perform well, as well as their combinations into portfolios. The next sections are dedicated to testing variants of CMA, in particular including quasi-random numbers and different parametrization.

B. *Tuning, and adding quasi-random numbers*

BI refers to initial step-size 40, whereas the initial step-size is 1 otherwise. QR refers to variants with quasi-random. MS refers to variants with step-size lower-bounded by 0.0001 instead of 0.01 for others. "+" refers to elitist strategies compared to the default "," strategy. Tables IV, V, VI, VII present the results for dimension 10 and 30 for the best, worst, median and average rank over the test cases. The best is usually better with quasi-random numbers, and the worst is worse. Means and medians are usually improved when using quasi-random numbers, though it is not always the case. In Section V, based on means and medians, we get a clear improvement with quasi-random numbers as all best methods use quasi-random mutations. BI usually provides an improvement, as well as the use of a "+" strategy.

IV. CEC 2015 CRITERIA

The CEC 2015 testbed uses an average between the mean and the best fitness, at the end of the budget of 1500 evaluations (in dimension 30) and 500 evaluations (in dimension 10), and also at half budget. The lowest errors we were able to obtain with one variant of CMA-ES (termed CMASPHI-QR in the following sections) are reproduced in tables VIII and IX. The corresponding complexities are shown in tables X and XI.

V. OTHER RESULTS

In this section, we present several results for all tested algorithms. We reproduced each run 10 times; the standard deviation provided in this section is the standard deviation of the score; i.e. they correspond to the variability of one run. Thanks to the averaging over 10 runs, our results are more significative.

Table XII synthesizes the results of different variants of CMA-ES. HI refers to 80 as an initial step-size. MI refers to 20 as an initial step-size. SP refers to small population size $\frac{1}{2}(3\log(N)+4)$ instead of $3\log(N)+4$. BP refers to big population size $2(3\log(N)+4)$.

Nelder-Mead variants are shown in table XIII. For Differential Evolution (Table XIV), the best performing one is a classical current to best with reasonably standard parametrization. PSO results are shown in table XV. Unless stated otherwise, the initial velocity is 1 and the maximum velocity MV is 1.5.

VI. CONCLUSION

We compared various algorithms on the CEC 2015 testbed. CMA and Nelder-Mead performed best. We validated quasi-random mutations for CMA, i.e. DCMA usually (though not always) outperformed CMA. In particular, all algorithms ranked 1 were equipped with QR mutations. Our best final combination is a flavor of DCMA, including covariance matrix adaptation and quasi-random numbers. We did not include mirroring[21]. We did not investigate sophisticated memmetic algorithms on top of CMA. We did not experiments on dimensions other than those proposed in the Cec2015 expensive optimization testbed. Portfolios methods were tested, and validated for the total budget, but not for the mid-budget criterion.

REFERENCES

[1] Q. Chen, B. Liu, Q. Zhang, J. J. Liang, P. N. Suganthan, and B. Y. Qu, "Problem definition and evaluation criteria for cec 2015 special session and competition on bound constrained single-objective computationally expensive numerical optimization," Computational Intelligence Laboratory, Zhengzhou University, China and Nanyang Technological University, Singapore, Tech. Rep., 2014.

[2] P. Baudis and P. Posik, "Online black-box algorithm portfolios for continuous optimization." in *Proceedings of PPSN*, 2014, pp. 40–49.

[3] M.-L. Cauwet, J. Liu, and O. Teytaud, "Algorithm portfolios for noisy optimization: Compare solvers early," in *Proceedings of the 8th Conference on Learning and Intelligent Optimization*, 2014.

[4] O. Teytaud and S. Gelly, "Dcma: yet another derandomization in covariance-matrix-adaptation," in *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2007, pp. 955–963.

[5] L. Pulina and A. Tacchella, "A self-adaptive multi-engine solver for quantified boolean formulas," *Constraints*, vol. 14, no. 1, pp. 80–116, 2009.

[6] M. Gagliolo and J. Schmidhuber, "A neural network model for inter-problem adaptive online time allocation," in *15th International Conference on Artificial Neural Networks: Formal Models and Their Applications*. Springer, 2005, pp. 7–12.

[7] ——, "Learning dynamic algorithm portfolios," vol. 47, no. 3-4, 2006, pp. 295–328.

[8] M. El-Beltagy, P. Nair, and A. Keane, "Metamodeling techniques for evolutionary optimization of computationally expensive problems: Promises and limitations," in *Proc. of the 1999 Genetic and Evolutionary COnference (GECCO 1999)*, D. G. . al., Ed. Morgan Kaufmann, 1999, pp. 196–203.

[9] C. Poloni and V. Pediroda, "GA coupled with computationaly expensive simulations: tools to improve efficiency," in *Genetic Algorithms and Evolution Strategies in Engineering and Computer Sciences*. John Wiley, 1997, pp. 267–288.

[10] A. Booker, J. D. Jr., P. Frank, D. Serafini, V. Torczon, and M. Trosset, "A rigorous framework for optimization of expensive functions by surrogates," *Structural Optimization*, vol. 17, no. 1, pp. 1–13, 1999.

[11] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[12] J. Villemonteix, E. Vazquez, and E. Walter, "An informational approach to the global optimization of expensive-to-evaluate functions," *Journal of Global Optimization*, Submitted.

[13] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 11, no. 1, 2003.

[14] H.-G. Beyer, *The Theory of Evolution Strategies*, ser. Natural Computing Series. Springer, Heideberg, 2001.

[15] I. Rechenberg, *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution*. Stuttgart: Fromman-Holzboog Verlag, 1973.

[16] R. Storn and K. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *J. of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997. [Online]. Available: http://dx.doi.org/10.1023/A:1008202821328

[17] H.-G. Beyer and B. Sendhoff, "Covariance matrix adaptation revisited - the CMSA evolution strategy," in *Proceedings of PPSN*, G. Rudolph, T. Jansen, S. M. Lucas, C. Poloni, and N. Beume, Eds., 2008, pp. 123–132.

Dim 10, worst																	
	CMA/CMA	CMA/ CMA /CMA	NM	DE/ DE /DE	NM/ DE /PSO	CMA	NM/ NM /NM	DE	CMSA/ NM /PSO	CMSA/ DE /PSO	SAiso	PSO/ PSO /PSO	SA	CMSA/ CMSA /CMSA	SAcov	CMSA	PSO
1	1	3	6	9	10	2	4	11	5	7	8	13	16	14	17	12	15
2	10	5	3	4	2	7	1	6	14	9	15	16	17	11	12	13	8
3	1	2	3	6	10	4	9	7	5	14	8	13	15	11	12	17	16
4	2	10	7	3	1	16	6	5	9	11	15	14	4	17	13	8	12
5	11	12	1	6	4	16	2	7	13	3	17	14	5	15	10	8	9
6	1	4	8	9	10	6	5	3	2	11	7	16	13	15	14	12	17
7	6	2	7	9	4	5	1	10	3	14	8	11	12	15	16	13	17
8	3	4	1	6	9	2	7	10	5	12	8	14	17	11	15	16	13
9	9	13	17	2	5	1	16	7	14	11	10	4	6	12	3	15	8
10	2	9	1	6	3	4	10	5	17	13	15	8	12	7	14	11	16
11	1	2	3	9	5	4	7	10	6	12	17	11	8	15	16	13	14
12	5	4	6	2	1	12	3	10	16	8	9	11	15	17	13	14	7
13	5	2	3	6	9	4	1	12	10	13	8	7	17	11	15	14	16
14	6	1	4	5	2	7	8	3	12	13	10	16	15	11	9	14	17
15	4	3	7	1	11	2	16	9	12	8	5	13	10	6	14	15	17
avg	4.5 $\pm$ 0.88	5.1 $\pm$ 1	5.1 $\pm$ 1.1	5.5 $\pm$ 0.7	5.7 $\pm$ 0.95	6.1 $\pm$ 1.2	6.4 $\pm$ 1.3	7.7 $\pm$ 0.74	9.5 $\pm$ 1.3	11 $\pm$ 0.79	11 $\pm$ 1	12 $\pm$ 0.9	12 $\pm$ 1.2	13 $\pm$ 0.85	13 $\pm$ 0.9	13 $\pm$ 0.66	13 $\pm$ 0.97
unimodal	5.5	4	4.5	6.5	6	4.5	2.5	8.5	9.5	8	12	14	16	12	14	12	12
multimodal	4.3	5.2	5.2	5.4	5.7	6.4	7	7.5	9.5	11	11	12	11	13	13	13	14

Dim 30, worst																	
	NM/ NM /NM	NM	CMA	CMA/CMA	CMA/ CMA /CMA	NM/ DE /PSO	DE	DE/ DE /DE	CMSA/ NM /PSO	CMSA/ DE /PSO	SAiso	PSO	PSO/ PSO /PSO	CMSA	CMSA/ CMSA /CMSA	SA	SAcov
1	5	4	3	1	2	7	9	10	6	11	8	12	13	15	14	16	17
2	1	2	6	11	7	5	4	8	16	10	14	3	9	15	13	12	17
3	2	4	1	8	9	6	7	5	3	11	10	16	15	13	12	17	14
4	1	2	8	6	7	5	3	4	14	15	9	10	16	12	11	17	13
5	1	2	16	14	15	4	11	12	13	3	9	10	7	5	8	6	17
6	4	1	5	3	2	8	10	9	6	11	7	13	12	17	16	15	14
7	5	6	2	1	3	8	9	7	4	10	14	11	12	17	13	15	16
8	4	5	1	2	3	8	9	7	6	10	11	13	12	15	14	16	17
9	3	5	10	13	15	4	1	9	6	8	11	2	16	14	17	12	7
10	1	3	5	6	2	8	9	4	12	7	10	14	13	11	17	15	16
11	4	5	2	1	3	8	10	7	6	9	12	13	11	14	17	16	15
12	1	10	3	9	4	2	5	7	8	6	14	12	11	17	15	13	16
13	1	3	5	2	6	10	11	9	4	8	7	14	12	13	15	17	16
14	5	7	4	1	2	8	9	10	6	3	13	11	12	16	15	17	14
15	4	5	1	2	3	6	8	7	11	9	10	14	17	12	13	15	16
avg	2.8 $\pm$ 0.44	4.3 $\pm$ 0.6	4.8 $\pm$ 1	5.3 $\pm$ 1.2	5.5 $\pm$ 1.1	6.5 $\pm$ 0.55	7.7 $\pm$ 0.78	7.7 $\pm$ 0.58	8.1 $\pm$ 1	8.7 $\pm$ 0.81	11 $\pm$ 0.62	11 $\pm$ 1	13 $\pm$ 0.69	14 $\pm$ 0.8	14 $\pm$ 0.64	15 $\pm$ 0.75	15 $\pm$ 0.66
unimodal	3	3	4.5	6	4.5	6	6.5	9	11	10	11	7.5	11	15	14	14	17
multimodal	2.8	4.5	4.8	5.2	5.7	6.5	7.8	7.5	7.6	8.5	11	12	13	14	14	15	15

TABLE I: Ranks of each algorithm for each test case, in dimension 10 and 30, for criteria worst.

[18] J. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal 7*, pp. 308–311, 1965.

[19] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[20] Y. Shi and R. C. Eberhart, "A Modified Particle Swarm Optimizer," in *Proceedings of IEEE International Conference on Evolutionary Computation*. Washington, DC, USA: IEEE Computer Society, May 1998, pp. 69–73.

[21] S. Gelly, J. Mary, and O. Teytaud, "On the ultimate convergence rates for isotropic algorithms and the best choices among various forms of isotropy," in 10^{th} *International Conference on Parallel Problem Solving from Nature (PPSN 2006)*, 2006.

Dim 10, median																	
	CMA/CMA	NM/ NM /NM	CMA/ CMA /CMA	NM	DE/ DE /DE	NM/ DE /PSO	CMA	DE	CMSA/ NM /PSO	CMSA/ DE /PSO	SAiso	SA	PSO/ PSO /PSO	CMSA/ CMSA /CMSA	SAcov	CMSA	PSO
1	1	2	4	6	9	10	3	11	5	7	8	16	13	14	17	12	15
2	13	1	6	3	4	2	9	7	16	12	11	5	17	14	8	15	10
3	3	9	2	4	6	10	1	7	5	14	8	15	13	11	12	17	16
4	3	2	10	7	4	1	16	6	9	11	15	5	14	17	13	8	12
5	1	3	13	2	7	5	16	10	14	4	17	6	15	9	12	8	11
6	1	3	5	8	9	10	6	4	2	11	7	13	16	15	14	12	17
7	6	1	2	7	9	4	5	10	3	14	8	12	11	15	16	13	17
8	4	3	5	1	7	9	2	10	6	12	8	17	14	11	15	16	13
9	9	16	13	17	2	5	1	7	14	11	10	6	4	12	3	15	8
10	2	11	10	1	7	4	5	6	3	14	16	13	9	8	15	12	17
11	2	4	1	5	10	6	3	8	7	14	17	9	12	11	13	15	16
12	5	3	4	6	2	1	13	10	11	8	9	16	12	17	14	15	7
13	5	1	2	3	6	9	4	12	10	13	8	17	7	11	15	14	16
14	6	8	1	4	5	2	7	3	12	13	10	15	16	11	9	14	17
15	5	7	3	11	2	13	4	10	15	1	8	12	6	9	14	16	17
avg	4.4 $\pm$ 0.85	4.9 $\pm$ 1.1	5.4 $\pm$ 1.1	5.7 $\pm$ 1.1	5.9 $\pm$ 0.7	6.1 $\pm$ 0.99	6.3 $\pm$ 1.3	8.1 $\pm$ 0.68	8.8 $\pm$ 1.2	11 $\pm$ 1	11 $\pm$ 0.94	12 $\pm$ 1.2	12 $\pm$ 1	12 $\pm$ 0.73	13 $\pm$ 0.93	13 $\pm$ 0.7	14 $\pm$ 0.91
unimodal	7	1.5	5	4.5	6.5	6	6	9	10	9.5	9.5	10	15	14	12	14	12
multimodal	4	5.5	5.5	5.8	5.8	6.1	6.4	7.9	8.5	11	11	12	11	12	13	13	14

Dim 30, median																	
	NM/ NM /NM	NM	CMA/CMA	CMA	CMA/ CMA /CMA	NM/ DE /PSO	CMSA/ NM /PSO	DE	DE/ DE /DE	CMSA/ DE /PSO	SAiso	PSO	PSO/ PSO /PSO	CMSA	CMSA/ CMSA /CMSA	SA	SAcov
1	6	4	1	3	2	7	5	11	10	9	8	12	13	15	14	16	17
2	2	1	12	7	6	5	11	4	8	10	15	3	9	16	14	13	17
3	4	7	3	1	2	5	6	9	8	10	15	13	16	12	17	11	14
4	1	3	9	2	8	4	7	5	6	15	10	11	16	13	12	17	14
5	1	3	16	15	17	4	2	10	5	6	12	8	7	13	14	9	11
6	4	1	3	6	2	5	7	10	9	11	8	13	12	17	16	15	14
7	3	7	2	4	5	1	6	9	8	10	14	11	12	15	13	16	17
8	4	6	3	2	1	8	5	9	7	10	11	14	13	16	15	17	12
9	3	6	9	13	16	4	7	1	12	10	11	2	5	15	17	14	8
10	1	2	5	4	3	7	12	9	6	8	10	15	13	11	17	14	16
11	5	4	3	1	2	9	6	10	7	8	12	13	14	16	11	15	17
12	4	2	1	7	5	3	10	8	9	6	13	11	12	15	16	14	17
13	2	1	3	5	6	8	4	11	10	9	7	14	12	13	15	17	16
14	3	7	2	6	1	8	4	9	11	5	10	12	13	16	15	17	14
15	4	5	2	1	3	6	7	9	10	11	8	16	17	13	12	14	15
avg	3.1 $\pm$ 0.39	3.9 $\pm$ 0.59	4.9 $\pm$ 1.2	5.1 $\pm$ 1.1	5.3 $\pm$ 1.3	5.6 $\pm$ 0.58	6.6 $\pm$ 0.7	8.3 $\pm$ 0.72	8.4 $\pm$ 0.51	9.2 $\pm$ 0.63	11 $\pm$ 0.67	11 $\pm$ 1	12 $\pm$ 0.84	14 $\pm$ 0.46	15 $\pm$ 0.5	15 $\pm$ 0.6	15 $\pm$ 0.68
unimodal	4	2.5	6.5	5	4	6	8	7.5	9	9.5	12	7.5	11	16	14	14	17
multimodal	3	4.2	4.7	5.2	5.5	5.5	6.4	8.4	8.3	9.2	11	12	12	14	15	15	14

TABLE II: Ranks of each algorithm for each test case, in dimension 10 and 30, for criteria median.

Dim 10, mean																	
	CMA/CMA	CMA/ CMA /CMA	NM	NM/ NM /NM	NM/ DE /PSO	DE/ DE /DE	CMA	DE	CMSA/ NM /PSO	CMSA/ DE /PSO	SAiso	PSO/ PSO /PSO	CMSA/ CMSA /CMSA	SA	SAcov	CMSA	PSO
1	1	3	5	4	10	9	2	11	6	7	8	13	14	16	17	12	15
2	11	5	3	1	2	4	8	6	15	10	14	17	12	16	9	13	7
3	3	2	4	10	9	6	1	7	5	14	8	13	11	15	12	17	16
4	3	10	7	2	1	4	16	6	9	11	15	14	17	5	13	8	12
5	4	13	1	2	5	7	15	10	14	3	17	16	11	6	12	9	8
6	1	5	8	4	10	9	6	3	2	11	7	16	15	13	14	12	17
7	6	2	7	1	4	9	5	10	3	14	8	11	15	12	16	13	17
8	3	4	1	5	9	7	2	10	6	12	8	14	11	17	15	16	13
9	9	13	17	16	2	3	1	7	14	11	10	5	12	6	4	15	8
10	2	9	1	10	3	6	4	5	13	14	16	8	7	12	15	11	17
11	1	2	5	4	6	10	3	8	7	12	17	11	13	9	15	14	16
12	5	4	6	2	1	3	12	10	13	8	9	11	17	16	14	15	7
13	4	2	3	1	9	6	5	12	10	13	8	7	11	17	15	14	16
14	6	2	4	8	1	5	7	3	12	13	10	16	11	15	9	14	17
15	5	3	9	11	12	1	4	10	14	2	7	6	8	13	15	16	17
avg	4.3 ± 0.75	5.3 ± 1	5.4 ± 1.1	5.4 ± 1.2	5.6 ± 1	5.9 ± 0.68	6.1 ± 1.2	7.9 ± 0.74	9.5 ± 1.1	10 ± 0.98	11 ± 0.99	12 ± 1	12 ± 0.75	13 ± 1.1	13 ± 0.88	13 ± 0.66	14 ± 1
unimodal	6	4	4	2.5	6	6.5	5	8.5	10	8.5	11	15	13	16	13	12	11
multimodal	4	5.5	5.6	5.8	5.5	5.8	6.2	7.8	9.4	11	11	11	12	12	13	13	14

Dim 30, mean																	
	NM/ NM /NM	NM	CMA	CMA/ CMA /CMA	CMA/CMA	NM/ DE /PSO	CMSA/ NM /PSO	DE/ DE /DE	DE	CMSA/ DE /PSO	PSO	SAiso	PSO/ PSO /PSO	CMSA	CMSA/ CMSA /CMSA	SAcov	SA
1	5	4	3	2	1	7	6	9	11	10	12	8	13	15	14	17	16
2	2	1	6	7	11	5	12	8	4	10	3	15	9	16	14	17	13
3	4	7	1	2	3	6	5	8	9	10	15	14	16	11	17	12	13
4	1	2	7	6	8	4	9	5	3	15	11	10	16	13	12	14	17
5	1	2	15	16	17	3	4	5	11	8	6	13	7	9	10	14	12
6	4	1	5	2	3	7	6	9	11	10	13	8	12	16	17	14	15
7	5	6	3	2	1	7	4	8	9	10	11	14	12	15	13	17	16
8	4	5	1	3	2	8	6	7	9	10	13	11	12	16	14	15	17
9	2	5	10	15	13	4	6	11	1	9	3	12	7	17	14	8	16
10	1	2	4	3	5	7	12	6	9	8	13	10	14	11	17	16	15
11	5	4	2	1	3	8	6	7	10	9	11	13	12	15	14	16	17
12	4	3	8	5	1	2	7	9	10	6	11	13	12	17	15	16	14
13	2	1	5	6	3	8	4	10	11	9	14	7	12	13	15	16	17
14	5	7	3	1	2	8	6	11	9	4	12	10	13	15	16	14	17
15	4	5	1	3	2	6	7	10	9	11	16	8	17	12	13	15	14
avg	3.3 ± 0.41	3.7 ± 0.55	4.9 ± 1	4.9 ± 1.2	5 ± 1.3	6 ± 0.51	6.7 ± 0.65	8.2 ± 0.5	8.4 ± 0.81	9.3 ± 0.62	11 ± 1	11 ± 0.67	12 ± 0.76	14 ± 0.63	14 ± 0.5	15 ± 0.61	15 ± 0.44
unimodal	3.5	2.5	4.5	4.5	6	6	9	8.5	7.5	10	7.5	12	11	16	14	17	14
multimodal	3.2	3.8	5	5	4.8	6	6.3	8.2	8.5	9.2	11	11	12	14	14	14	15

TABLE III: Ranks of each algorithm for each test case, in dimension 10 and 30, for criteria mean.

Dim 10, best								
	CMABIQR	CMA+QR	CMA+BISMQR	CMABIMSQR	CMABI	CMABIMS	CMA+	CMABI+MS
1	2	1	3	4	7	6	5	8
2	2	1	3	4	5	7	8	6
3	4	1	5	6	2	8	7	3
4	4	1	5	6	2	7	8	3
5	2	1	3	4	6	5	8	7
6	1	4	2	3	6	5	8	7
7	2	1	3	4	6	8	5	7
8	1	4	2	3	5	6	7	8
9	2	1	3	4	7	6	5	8
10	1	5	2	3	4	6	7	8
11	1	4	2	3	7	6	5	8
12	1	4	2	3	6	5	8	7
13	1	4	2	3	5	6	7	8
14	2	1	3	4	6	5	8	7
15	3	1	4	5	7	2	6	8
avg	1.9 $\pm$ 0.27	2.3 $\pm$ 0.42	2.9 $\pm$ 0.27	3.9 $\pm$ 0.27	5.4 $\pm$ 0.42	5.9 $\pm$ 0.38	6.8 $\pm$ 0.33	6.9 $\pm$ 0.43
unimodal	2	1	3	4	6	6.5	6.5	7
multimodal	1.9	2.5	2.9	3.9	5.3	5.8	6.8	6.8

Dim 30, best								
	CMABIQR	CMA+QR	CMA+BISMQR	CMABIMSQR	CMABI	CMABIMS	CMABI+MS	CMA+
1	1	4	2	3	7	6	8	5
2	1	6	2	3	4	8	5	7
3	2	1	3	4	6	8	7	5
4	2	1	3	4	5	8	6	7
5	1	4	2	3	6	5	7	8
6	2	1	3	4	6	5	7	8
7	3	2	4	5	6	1	7	8
8	3	1	4	5	7	8	2	6
9	2	1	3	4	7	5	8	6
10	1	5	2	3	4	7	8	6
11	3	1	4	5	2	8	6	7
12	2	1	3	4	5	7	6	8
13	1	5	2	3	6	4	7	8
14	3	1	4	5	6	2	7	8
15	5	1	6	7	3	8	4	2
avg	2.1 $\pm$ 0.29	2.3 $\pm$ 0.48	3.1 $\pm$ 0.29	4.1 $\pm$ 0.29	5.3 $\pm$ 0.39	6 $\pm$ 0.59	6.3 $\pm$ 0.42	6.6 $\pm$ 0.43
unimodal	1	5	2	3	5.5	7	6.5	6
multimodal	2.3	1.9	3.3	4.3	5.3	5.8	6.3	6.7

TABLE IV: Ranks of each CMA variant for each test case, in dimension 30, for criterion best.

Dim 10, worst								
	CMABI	CMABIMS	CMA+	CMABI+MS	CMABIQR	CMA+BISMQR	CMABIMSQR	CMA+QR
1	6	1	2	7	3	4	5	8
2	1	3	7	2	4	5	6	8
3	1	4	5	2	6	7	8	3
4	1	3	4	2	6	7	8	5
5	2	1	7	3	4	5	6	8
6	2	1	4	3	6	7	8	5
7	4	3	1	2	5	6	7	8
8	1	2	3	4	6	7	8	5
9	3	2	1	4	5	6	7	8
10	3	1	2	4	5	6	7	8
11	2	1	4	3	5	6	7	8
12	2	1	3	4	5	6	7	8
13	2	7	1	3	4	5	6	8
14	2	1	4	3	5	6	7	8
15	2	4	1	3	5	6	7	8
avg	2.3 $\pm$ 0.34	2.3 $\pm$ 0.44	3.3 $\pm$ 0.52	3.3 $\pm$ 0.33	4.9 $\pm$ 0.23	5.9 $\pm$ 0.23	6.9 $\pm$ 0.23	7.1 $\pm$ 0.43
unimodal	3.5	2	4.5	4.5	3.5	4.5	5.5	8
multimodal	2.1	2.4	3.1	3.1	5.2	6.2	7.2	6.9

Dim 30, worst								
	CMABI	CMABI+MS	CMA+	CMABIMS	CMABIQR	CMA+QR	CMA+BISMQR	CMABIMSQR
1	7	8	2	6	3	1	4	5
2	2	4	1	3	5	8	6	7
3	3	4	1	2	5	8	6	7
4	1	2	3	4	5	8	6	7
5	1	2	3	4	5	8	6	7
6	3	4	1	2	6	5	7	8
7	1	2	3	4	5	8	6	7
8	4	1	2	8	5	3	6	7
9	2	3	4	1	6	5	7	8
10	1	3	4	2	6	5	7	8
11	1	2	3	4	5	8	6	7
12	1	4	3	2	5	8	6	7
13	3	2	4	1	5	8	6	7
14	1	3	8	2	5	4	6	7
15	2	3	7	8	4	1	5	6
avg	2.2 $\pm$ 0.43	3.1 $\pm$ 0.42	3.3 $\pm$ 0.52	3.5 $\pm$ 0.58	5 $\pm$ 0.2	5.9 $\pm$ 0.68	6 $\pm$ 0.2	7 $\pm$ 0.2
unimodal	4.5	6	1.5	4.5	4	4.5	5	6
multimodal	1.8	2.7	3.5	3.4	5.2	6.1	6.2	7.2

TABLE V: Ranks of each CMA variant for each test case, in dimension 30, for criterion worst.

Dim 10, median								
	CMABIQR	CMABIMS	CMA+BISMQR	CMABI	CMA+QR	CMA+	CMABIMSQR	CMABI+MS
1	1	6	2	7	4	5	3	8
2	3	7	4	1	6	8	5	2
3	5	8	6	3	1	2	7	4
4	4	3	5	1	7	8	6	2
5	4	1	5	2	7	8	6	3
6	1	5	2	6	4	8	3	7
7	3	8	4	6	2	1	5	7
8	4	2	5	1	8	3	6	7
9	3	2	4	6	8	1	5	7
10	3	1	4	6	8	2	5	7
11	2	5	3	7	6	1	4	8
12	3	1	4	6	2	8	5	7
13	1	8	2	4	6	5	3	7
14	4	1	5	2	3	8	6	7
15	4	1	5	8	2	7	6	3
avg	3 $\pm$ 0.32	3.9 $\pm$ 0.75	4 $\pm$ 0.32	4.4 $\pm$ 0.65	4.9 $\pm$ 0.64	5 $\pm$ 0.78	5 $\pm$ 0.32	5.7 $\pm$ 0.57
unimodal	2	6.5	3	4	5	6.5	4	5
multimodal	3.2	3.5	4.2	4.5	4.9	4.8	5.2	5.8

Dim 30, median								
	CMABIQR	CMA+BISMQR	CMA+QR	CMABI	CMABIMS	CMABIMSQR	CMABI+MS	CMA+
1	1	2	4	8	5	3	7	6
2	1	2	8	5	7	3	6	4
3	5	6	1	4	8	7	3	2
4	3	4	7	1	8	5	6	2
5	2	3	5	6	1	4	7	8
6	3	4	2	7	1	5	8	6
7	4	5	3	7	1	6	2	8
8	3	4	2	6	8	5	1	7
9	5	6	1	4	2	7	3	8
10	3	4	8	1	6	5	7	2
11	2	3	5	6	8	4	7	1
12	2	3	5	6	7	4	1	8
13	3	4	6	2	1	5	7	8
14	5	6	3	2	1	7	4	8
15	2	3	1	7	8	4	6	5
avg	2.9 $\pm$ 0.34	3.9 $\pm$ 0.34	4.1 $\pm$ 0.64	4.8 $\pm$ 0.6	4.8 $\pm$ 0.82	4.9 $\pm$ 0.34	5 $\pm$ 0.62	5.5 $\pm$ 0.69
unimodal	1	2	6	6.5	6	3	6.5	5
multimodal	3.2	4.2	3.8	4.5	4.6	5.2	4.8	5.6

TABLE VI: Ranks of each CMA variant for each test case, in dimension 30, for criterion median.

Dim 10, mean								
	CMABIQR	CMABIMS	CMA+BISMQR	CMABI	CMABIMSQR	CMABI+MS	CMA+	CMA+QR
1	1	5	2	7	3	6	4	8
2	3	6	4	1	5	2	8	7
3	4	8	5	1	6	3	7	2
4	5	3	6	1	7	2	8	4
5	4	1	5	2	6	3	8	7
6	2	1	3	6	4	7	8	5
7	4	8	5	3	6	2	1	7
8	4	2	5	1	6	7	3	8
9	3	2	4	7	5	8	1	6
10	3	1	4	6	5	7	2	8
11	1	5	2	6	3	7	4	8
12	2	1	3	5	4	6	8	7
13	1	8	2	4	3	6	5	7
14	4	1	5	2	6	7	8	3
15	3	1	4	8	5	7	6	2
avg	2.9 $\pm$ 0.33	3.5 $\pm$ 0.74	3.9 $\pm$ 0.33	4 $\pm$ 0.66	4.9 $\pm$ 0.33	5.3 $\pm$ 0.57	5.4 $\pm$ 0.7	5.9 $\pm$ 0.56
unimodal	2	5.5	3	4	4	4	6	7.5
multimodal	3.1	3.2	4.1	4	5.1	5.5	5.3	5.7

Dim 30, mean								
	CMABIQR	CMA+BISMQR	CMABI	CMA+QR	CMABIMS	CMABI+MS	CMABIMSQR	CMA+
1	1	2	8	4	6	7	3	5
2	2	3	1	8	7	6	4	5
3	4	5	3	2	8	7	6	1
4	4	5	1	7	8	3	6	2
5	2	3	5	7	1	6	4	8
6	3	4	6	2	1	8	5	7
7	4	5	7	3	1	2	6	8
8	3	4	7	2	8	1	5	6
9	4	5	7	1	2	3	6	8
10	4	5	1	8	3	7	6	2
11	2	3	1	8	7	5	4	6
12	4	5	1	3	7	2	6	8
13	2	3	6	5	1	7	4	8
14	5	6	2	3	1	4	7	8
15	2	3	7	1	8	6	4	5
avg	3.1 $\pm$ 0.3	4.1 $\pm$ 0.3	4.2 $\pm$ 0.73	4.3 $\pm$ 0.69	4.6 $\pm$ 0.82	4.9 $\pm$ 0.58	5.1 $\pm$ 0.3	5.8 $\pm$ 0.63
unimodal	1.5	2.5	4.5	6	6.5	6.5	3.5	5
multimodal	3.3	4.3	4.2	4	4.3	4.7	5.3	5.9

TABLE VII: Ranks of each CMA variant for each test case, in dimension 30, for criterion mean.

F	Best	Worst	Median	Mean	Std
1	3066.821720	27350.414967	10303.331393	13532.288748	8015.631303
2	17946.244786	75601.839181	38585.782508	39525.522486	15424.252191
3	1.010692	15.066042	6.772117	7.342122	4.307489
4	130.551411	2754.489295	2213.214380	1919.328740	798.520928
5	0.428649	4.694434	3.378933	3.178128	1.239418
6	0.257797	0.816413	0.500261	0.494573	0.153775
7	0.288968	1.247096	0.495755	0.590027	0.302133
8	1.621837	6.750625	4.885070	4.552976	1.573973
9	3.353918	4.595982	4.222199	4.156896	0.340649
10	11346.738112	2450865.764312	263107.148179	556046.674992	730726.425838
11	2.187530	40.353662	6.419068	8.149008	7.903810
12	31.308448	610.640458	307.249692	322.513038	153.876607
13	316.642541	333.405878	320.124508	321.868712	4.217949
14	186.334995	210.410137	202.610188	201.771257	6.009089
15	10.511656	528.958620	408.019290	340.349673	170.162921

TABLE VIII: Errors obtained by CMASPHI-QR in 10D

F	Best	Worst	Median	Mean	Std
1	4410.546684	2433205.172072	9909.121145	203114.035298	611688.572924
2	73838.778110	155921.584174	118489.287353	117191.061224	21987.789028
3	13.348001	37.062026	20.819096	21.305606	5.062646
4	2553.626647	9575.708781	8591.667368	7588.016926	2185.679036
5	4.662556	6.673121	5.284304	5.437349	0.591955
6	0.373039	1.045966	0.659812	0.685387	0.235695
7	0.323589	1.313365	0.535025	0.636289	0.286018
8	6.749165	56.118232	22.244325	22.911469	10.927353
9	13.241557	14.225287	13.976680	13.848334	0.303010
10	610222.482957	8220937.807046	2549366.836787	2774403.117855	2212879.888269
11	17.995147	88.531156	22.161005	27.828859	19.823562
12	224.026689	1078.687135	640.246067	671.996207	282.541119
13	354.170628	400.510809	376.774443	377.075234	11.912947
14	218.364928	271.850932	243.197289	245.710839	15.669614
15	600.342406	887.127952	771.879600	758.132119	79.104475

TABLE IX: Errors obtained by CMASPHI-QR in 30D

Func.	$\hat{T_1}/T_0$
1	5.20045e-01
2	2.61135e-01
3	1.05740e+00
4	2.81510e-01
5	5.36625e-01
6	3.41910e-01
7	3.59610e-01
8	3.36335e-01
9	2.66570e-01
10	2.80205e-01
11	4.34245e-01
12	3.10260e-01
13	2.94360e-01
14	3.13660e-01
15	1.13885e+00

TABLE X: Computational Complexity for best results in 10D

Func.	$\hat{T_1}/T_0$
1	1.25175e+00
2	1.17270e+00
3	8.35025e+00
4	1.27570e+00
5	3.27215e+00
6	1.22645e+00
7	1.24210e+00
8	1.29370e+00
9	1.15715e+00
10	1.46540e+00
11	2.71990e+00
12	1.56960e+00
13	1.54560e+00
14	1.67210e+00
15	8.83170e+00

TABLE XI: Computational Complexity for best results in 30D

CMA Variants	Mean	STD	Median	Rank
CMA+	5.78e8	1.60e8	5.34e8	20
CMA+QR	4.54e8	2.90e7	4.59e8	15
CMABI+	5.99e8	1.81e8	5.41e8	22
CMABIQR	4.52e8	2.81e7	4.46e8	14
CMABI	6.17e8	1.94e8	5.80e8	24
CMABISS+QR	4.54e8	2.90e7	4.59e8	15
CMABISS+	6.06e8	1.87e8	5.38e8	23
CMABISSQR	4.51e8	2.70e7	4.49e8	13
CMABISS	5.58e8	1.74e8	5.36e8	19
CMAMI+	5.79e8	1.73e8	5.98e8	21
CMAMI+QR	4.55e8	2.83e7	4.59e8	18
CMASPBI+	7.20e7	1.98e7	7.69e7	10
CMASPBI+QR	5.54e7	8.35e6	5.38e7	6
CMABPBI+	5.39e9	6.23e8	5.40e9	27
CMABPBI+QR	4.30e9	1.98e8	4.24e9	25
CMABPBI	5.72e9	1.20e9	5.62e9	28
CMABPBIQR	4.30e9	1.98e8	4.24e9	25
CMASPBI	8.08e7	2.52e7	7.66e7	12
CMASPBIQR	5.21e7	3.47e6	5.20e7	1
CMASPBISS+	7.39e7	2.66e7	6.42e7	11
CMASPBISS+QR	5.24e7	3.55e6	5.26e7	5
CMASPBISS	6.57e7	1.99e7	6.97e7	8
CMASPBISS+QR	5.21e7	3.47e6	5.20e7	1
CMASPMI+	6.96e7	1.56e7	6.76e7	9
CMASPMI+QR	5.21e7	3.47e6	5.20e7	1
CMASPHI+	6.53e7	1.80e7	6.34e7	7
CMASPHI+QR	5.21e7	3.47e6	5.20e7	1

TABLE XII: Results of different variants of CMA-ES

NM Variants	Mean	STD	Median	Rank
$\alpha = 1, \gamma = 2, \rho = -.5, \sigma = .5$	1.11e10	5.16e9	1.00e10	2
$\alpha = .8, \gamma = 2, \rho = -.5, \sigma = .5$	3.27e10	1.16e10	2.98e10	5
$\alpha = 1, \gamma = 2, \rho = -.75, \sigma = .5$	1.13e10	6.51e9	9.91e9	3
$\alpha = 1, \gamma = 2, \rho = -.5, \sigma = .75$	1.06e10	5.41e9	9.72e9	1
$\alpha = 1.5, \gamma = 3, \rho = -.25, \sigma = .25$	3.55e10	1.28e10	3.26e10	6
$\alpha = .5, \gamma = 1.5, \rho = -.75, \sigma = .75$	9.84e10	1.02e10	1.00e11	7

TABLE XIII: Results of different variants of Nelder-Mead

DE Variants	Mean	STD	Median	Rank
$DE/curr-to-best/1$ $f_1 = .8, f_2 = .8, cr = .5$	7.17e10	8.69e9	7.14e10	8
$DE/curr-to-best/1$ $f_1 = 1, f_2 = 1, cr = .8$	1.34e11	2.52e10	1.38e11	9
$DE/curr-to-best/1$ $f_1 = 1, f_2 = 1, cr = .5$	6.72e10	1.29e10	6.96e10	1
$DE/curr-to-best/1$ $f_1 = .5, f_2 = .8, cr = .5$	7.02e10	1.24e10	7.01e10	5
$DE/rand-to-best/1$ $f_1 = 1, f_2 = 1, cr = .5$	7.14e10	1.12e10	6.99e10	6
$DE/best/1$ $f_1 = 1, f_2 = 1, cr = .5$	6.95e10	9.39e9	6.97e10	4
$DE/best/1$ $f_1 = 1, f_2 = 1, cr = .5$	6.94e10	9.55e9	6.97e10	3
$DE/best/1$ $f_1 = 1, f_2 = 1, cr = .5$	6.94e10	9.54e9	6.97e10	2

TABLE XIV: Results of different variants of Differential Evolution

PSO Variants	Mean	STD	Median	Rank
$\mu = 30, neighb = 10, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$	1.46e11	2.33e10	1.44e11	12
$\mu = 30, neighb = 5, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$	1.50e11	1.60e10	1.48e11	13
$\mu = 15, neighb = 5, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$ 1702	1.45e11	1.43e10	1.41e11	11
$MV = 2.5, \mu = 30, neighb = 15, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$	1.13e11	1.57e10	1.13e11	8
$MV = 5, \mu = 15, neighb = 5, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$	8.48e10	1.63e10	7.78e10	5
$MV = 2.5, \mu = 30, neighb = 10, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$ 1705	1.11e11	1.54e10	1.11e11	6
$MV = 2.5, \mu = 30, neighb = 10, \omega = 1/(2\log(2)),$ $\phi_p = 1.5 + \log(2), \phi_g = 1.5 + \log(2)$	1.14e11	1.55e10	1.16e11	9
$MV = 1.5, \mu = 30, neighb = 10, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$	1.43e11	1.22e10	1.45e11	10
$MV = 2.5, \mu = 30, neighb = 10, \omega = 1/(2\log(2)),$ $\phi_p = 1.5 + \log(2), \phi_g = 1.5 + \log(2)$ 1708	1.11e11	1.13e10	1.11e11	7
$MV = 10, \mu = 30, neighb = 5, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$	5.63e10	8.16e9	5.77e10	3
$MV = 20, \mu = 30, neighb = 5, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$	4.45e10	1.34e10	4.60e10	1
$MV = 10, \mu = 30, neighb = 5, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$ 1711	5.89e10	1.27e10	5.50e10	4
$MV = 20, \mu = 30, neighb = 5, \omega = 1/(2\log(2)),$ $\phi_p = .5 + \log(2), \phi_g = .5 + \log(2)$	4.64e10	1.49e10	4.70e10	2

TABLE XV: Results of different variants of PSO