Benchmarking an algorithm for expensive high-dimensional objectives on the bbob and bbob-largescale testbeds

Zachary Hoffman, Steve Huntsman

Benchmarking an algorithm for expensive high-dimensional objectives on the bbob and bbob-largescale testbeds

Zachary Hoffman
Systems and Technology Research
Woburn, Massachusetts
zachary.hoffman@str.us

Steve Huntsman
Systems and Technology Research
Arlington, Virginia
steve.huntsman@str.us

ABSTRACT

We report benchmarks for a recently developed algorithm on the `bbob` and `bbob-largescale` benchmarking testbeds in COCO. This algorithm is designed for expensive high-dimensional multimodal objectives (such as arise in hyperparameter optimization or via simulations), and this regime introduces challenges for benchmarking. In particular, while the COCO experimental procedure yields evidence that this algorithm improves on the state of the art, the COCO framework also exhibits shortcomings for the very low evaluation budgets involved here. Consequently, we also report the results of *ad hoc* experiments that demonstrate a more obvious if less rigorous advantage for the algorithm over its nearest competition.

CCS CONCEPTS

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

KEYWORDS

Benchmarking, Black-box optimization

ACM Reference Format:
Zachary Hoffman and Steve Huntsman. 2022. Benchmarking an algorithm for expensive high-dimensional objectives on the `bbob` and `bbob-largescale` testbeds. In *Proceedings of the Genetic and Evolutionary Computation Conference 2022 (GECCO '22).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/123_4

1 INTRODUCTION

The paper [13] introduces an optimization algorithm designed to handle intrinsically high-dimensional ($D \gtrapprox 40$, and with all dimensions relevant) optimization problems for very expensive (≥ 1 minute/evaluation) multimodal objective functions. This regime includes problems of fundamental interest (e.g., optimizing the results of simulations) that push against fundamental limits of benchmarking tools: for example, high dimensionality and multimodality coupled with low function evaluation budgets preclude achieving global minima as a realistic goal. Meanwhile, while the algorithm in [13] is carefully designed with such problems in mind, its runtime overhead introduces an additional challenge: benchmarking itself becomes very expensive.

The *COmparing Continuous Optimizers* (COCO) platform [9] enables automated benchmarking and performance comparisons for optimization algorithms. In this paper, we benchmark the algorithm in [13]–hereinafter referred to as `EXPLO2` in the text and `EE` in the COCO plots–on the `bbob` [5, 10] and `bbob-largescale` [4] test suites, as well as performing a more *ad hoc* evaluation to address gaps in COCO for low function evaluation budgets.

2 ALGORITHM PRESENTATION

If d is the distance matrix of a finite subset of $\mathbb{R}^D$ and $t \geq 0$, then $Z_{jk} = \exp(-td_{jk})$ is a positive definite matrix [20], hence invertible. With this in mind, the corresponding *weighting* w is the solution of $Zw = 1$, where 1 denotes a vector of all ones; the *magnitude* of Z is $\sum_j w_j$ [15]. Magnitude is a very general scale-dependent notion of an effective number of points that encodes rich geometrical data [16]. Meanwhile, the components of a weighting meaningfully encode a notion of effective size per point.

Moreover, Z is a *radial basis function* (RBF) interpolation matrix [2] and the equation $Zw = 1$ amounts to defining a weighting w as the vector whose components are coefficients for interpolating the unit function. So if $\{x_j\}_{j=1}^n$ are points in $\mathbb{R}^D$ with distance matrix d and $y_j := f(x_j)$ for $f : \mathbb{R}^D \to \mathbb{R}$, then the RBF interpolation is $f(x) \approx yZ^{-1}\zeta(x)$, where $\zeta_k(x) := \exp(-t|x - x_k|)$ and we treat y and ζ respectively as row and column vectors.

`EXPLO2` constructs a surrogate on a box in $\mathbb{R}^D$ that convexly combines i) the differential magnitude due to a new point in relation to a dynamically curated subset of prior evaluation points, and ii) the RBF interpolation relative to those same points. These two functions respectively embody notions of exploration and exploitation, and over the course of the algorithm the surrogate shifts from the former to the latter. Successive evaluation points are determined by optimizing the surrogate with an inner large-scale algorithm (default: `fmincon`).

3 EXPERIMENTAL PROCEDURE

We ran `EXPLO2` with default settings as described in [13] and with $n_\| = 32$ parallel workers on the entire `bbob` suite for $25 \times D$ evaluations according to [12]. We also ran `EXPLO2` in the same way on the multimodal functions $f_{15}, \dots, f_{19}$ in the `bbob-largescale` suite for $2 \times D$ evaluations. [1] The limited budget and function selection were due to runtime overhead, as discussed in the sequel.

*This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

GECCO '22, July 9–13, 2022, Boston, MA, USA

ACM ISBN 123-4567-89-101/11/21.
https://doi.org/10.1145/123_4

[1] We also ran `EXPLO2` in the same way on the weakly structured multimodal functions $f_{20}, \dots, f_{24}$ in the `bbob-largescale` suite, but with degenerate results.

As [13] details, a representative set of algorithms to benchmark against for low-dimensional, expensive, multimodal functions is NEWUOA, MCS, GLOBAL, SMAC-BBOB, and *-CMA-ES with $* \in \{$lmm, DTS, lq$\}$. Of these, only NEWUOA is well-suited to problems with hundreds of dimensions. Indeed, most of these algorithms have not even been benchmarked for dimension 40 in COCO as of this writing, and even NEWUOA eventually scales poorly with dimension [19]. Thus it is also important to perform large-scale benchmarking of EXPLO2, and to compare its performance to fmincon, which EXPLO2 is a wrapper for and is itself a large-scale algorithm.

This experimental procedure was useful, but still insufficient to adequately characterize the regime for which EXPLO2 was designed and where it exhibits advantage over competing algorithms, viz. very expensive high-dimensional multimodal objectives. In particular, this procedure did not yield plots in COCO that readily conveyed meaningful information (e.g., an "expensive plot" option does not appear to be available for the bbob-largescale suite). We therefore also performed a more *ad hoc* comparison of EXPLO2 to VD-CMA-ES [1], the best performing large-scale algorithm for structured multimodal functions in [23]. As Figures 10-13 show, EXPLO2 performs better with respect to evaluation counts, regardless of parallelism. However, it also incurs a very large runtime overhead.

4 CPU TIMING

In fact, the CPU timing of EXPLO2 is orders of magnitude higher than competing algorithms, as it wraps a large amount of dense linear algebra (albeit with carefully controlled runtime, and also with prospects for eventually achieving sparsity) around an inner optimizer. Therefore, EXPLO2 is only indicated instead of *-CMA-ES for objectives whose evaluation requires on the order of minutes or more. For example, the bbob data presented here took roughly a week to generate; the bbob-largescale data was similarly very time-consuming. For this reason, precise timing measurements were largely pointless, and high function evaluation budgets were impractical.

5 RESULTS

Results from experiments according to [12] and [8] on the benchmark functions given in [4, 5] are presented in Figures 1, 2-3, and 4-8; see also Table 1. [2] The experiments were performed and plots produced with COCO [9], version 2.4. Additionally, our *ad hoc* experiments with VD-CMA-ES are presented in Figures 10-13.

The **expected runtime (ERT)**, used in Table 1, depends on a given target precision, $I_{\text{target}} = f_{\text{opt}} + \Delta f$, 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 [12, 17]. **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 f_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.

Because of the runtime overhead of EXPLO2, our experiments used a fixed budget of 25 function evaluations per dimension (i.e., $25 \times \text{DIM}$ function evaluations in total). [3] As [12] points out, algorithms are only comparable up to the smallest budget given to any of them, corresponding in this case to $\log_{10} 25 \approx 1.40$ on the horizontal axis for Figures 2-3 and 4-7, and to $\log_{10} 2 \approx 0.3$ for Figure 8. This corresponds in our case exactly to the location of crosses ($\times$), which indicate where bootstrapping of experimental data begins to estimate results for larger numbers of function evaluations. At the same time, we used $n_\parallel = 32$ parallel workers, [4] and $\log_{10}(25/32) \approx -0.107$. Thus allowing for parallel resources (see §5.1) suggests comparing EXPLO2 at the value $\log_{10} 25$ on the horizontal axes with the other algorithms at $\log_{10}(25/32)$, except for *-CMA-ES, which is parallelizable. [5] [6]

To summarize, as the dimension of structured multimodal problems grows, EXPLO2 outperforms NEWUOA, particularly when taking parallelism into account; meanwhile, the quadratic scaling with dimension of non-large-scale *-CMA-ES algorithms puts them at an increasing disadvantage since EXPLO2 has essentially no runtime dependence on dimension other than through its inner solver, which here is the default large-scale interior-point fmincon algorithm. Finally, on structured multimodal functions, EXPLO2 outperforms VD-CMA-ES, which is otherwise the best-performing large-scale algorithm that we are aware of.

5.1 Parallel performance

While EXPLO2 is parallelized (at marginal cost to performance), to the best of our knowledge no competing technique is except for *-CMA-ES. Indeed, as [6] points out, most surrogate-based derivative-free optimizers–including NEWUOA–require sequential function evaluations. Though parallel algorithms exist [7, 18, 25], these are relatively few in number outside the context of Bayesian optimization (which is unsuitable for high-dimensional problems), and parallel techniques appropriate for high-dimensional problems have been considered in our design and/or benchmarking.

With this in mind, since our experiments use $n_\parallel = 32$, exceeding our per-dimension evaluation budget of 25, it is obvious that EXPLO2 can outperform any of the competing algorithms considered here on the number of rounds of parallel function evaluations, except for *-CMA-ES, which becomes (depending on the variant employed) ill-suited for use or less performant than EXPLO2 in high dimensions.

[2] In an appendix of [13], we also produced fixed-budget plots of cumulative best values using the same data (and substituting, e.g. DTS-CMA-ES for lmm-CMA-ES) via the web interface of IOHanalyzer [3] at https://iohanalyzer.liacs.nl/, but these plots produced neither additional nor conflicting insights.

[3] While [22] points out how an anytime benchmark of a budget-dependent algorithm such as EXPLO2 can be performed with linear overhead, the cost-to-benefit ratio in our case was still prohibitive. In any event, this would not have made larger budgets any easier (or much more relevant) to obtain.

[4] NB. For benchmarking, it was necessary to simulate parallelism, i.e., we replaced parfor loops in MATLAB code with ordinary for loops.

[5] While *-CMA-ES is easily parallelizable [11, 14], the quadratic scaling of non-large-scale variants with dimension creates a serious disadvantage for high-dimensional problems. In high dimensions it is appropriate to compare EXPLO2 to large-scale variants of CMA-ES [23, 24], and our *ad hoc* experiments in this regard yielded good results.

[6] Also, in an experiment shown in an appendix of [13], we found that EXPLO2 outperformed differential evolution and had performance almost indistinguishable from CMA-ES on the mixed-integer version of f_{15} [21] in dimension $D = 20$ (the only function/dimension pair we tried among the mixed-integer suite [21]).

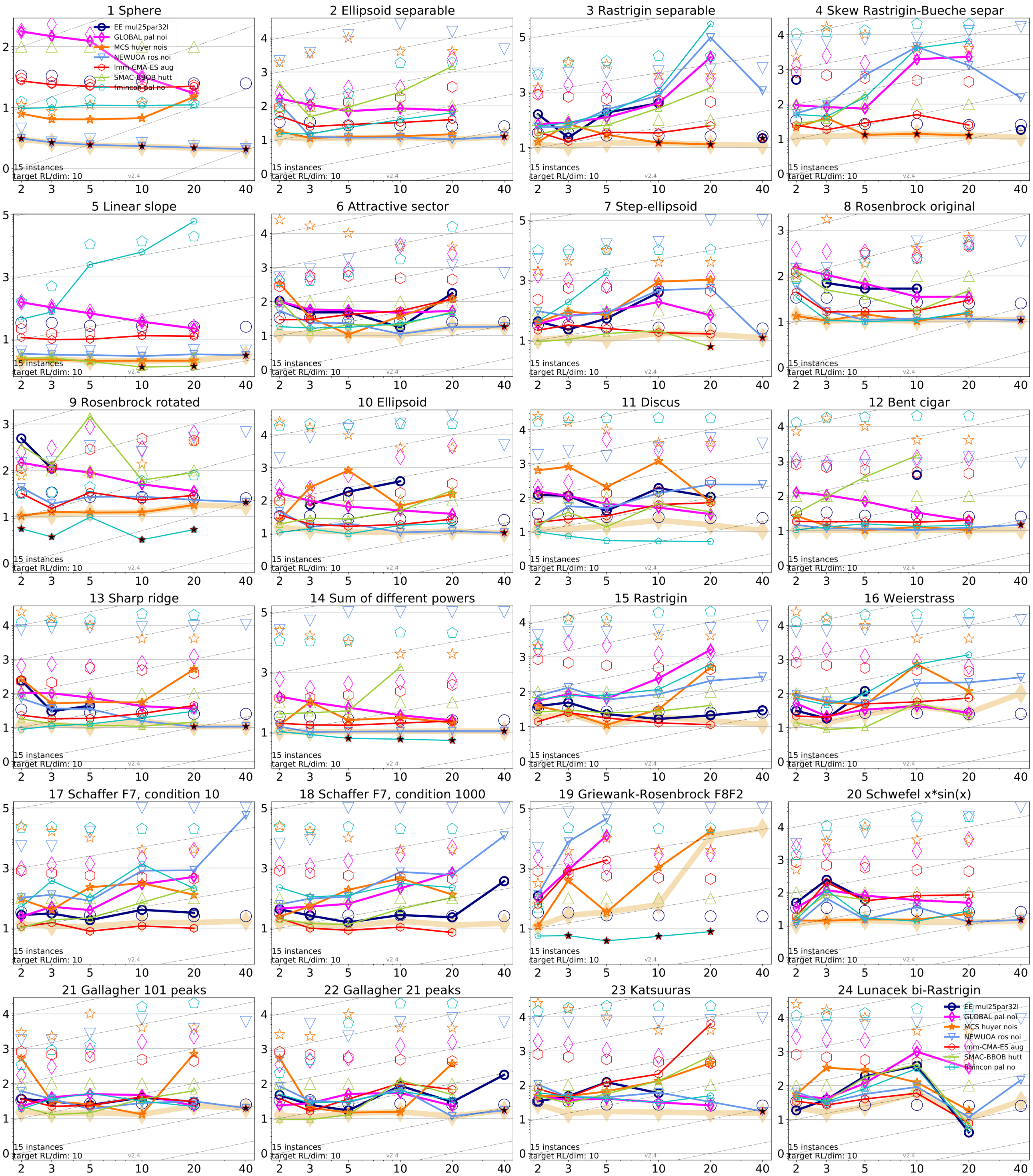

Figure 1: Expected running time (ERT in number of f-evaluations as $\log_{10}$ value) divided by dimension versus dimension. The target function value is chosen such that the best algorithm from BBOB 2009 just failed to achieve an ERT of $10 \times \mathrm{DIM}$. Different symbols correspond to different 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: ○: EE mul25par32lamLin, ◇: GLOBAL pal noiseless, ⋆: MCS huyer noiseless, ▽: NEWUOA ros noiseless, ⬡: SMAC-BBOB hutter noiseless, △: fmincon pal noiseless, ⬠: lmm-CMA-ES auger noiseless.

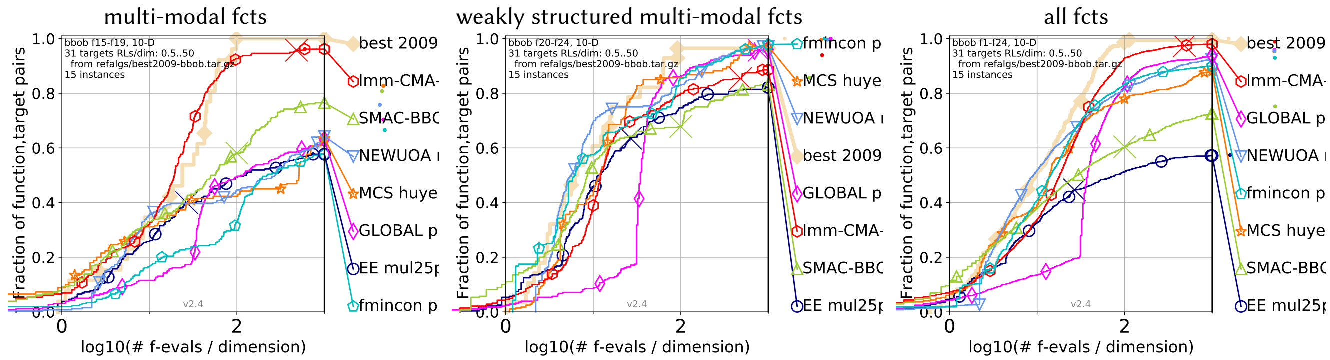

Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for all functions and subgroups in 10-D. The targets are chosen from $10^{[-8..2]}$ such that the best algorithm from BBOB 2009 just not reached them within a given budget of $k \times$ DIM, with 31 different values of k chosen equidistant in logscale within the interval $\{0.5, \ldots, 50\}$. As reference algorithm, the best algorithm from BBOB 2009 is shown as light thick line with diamond markers.

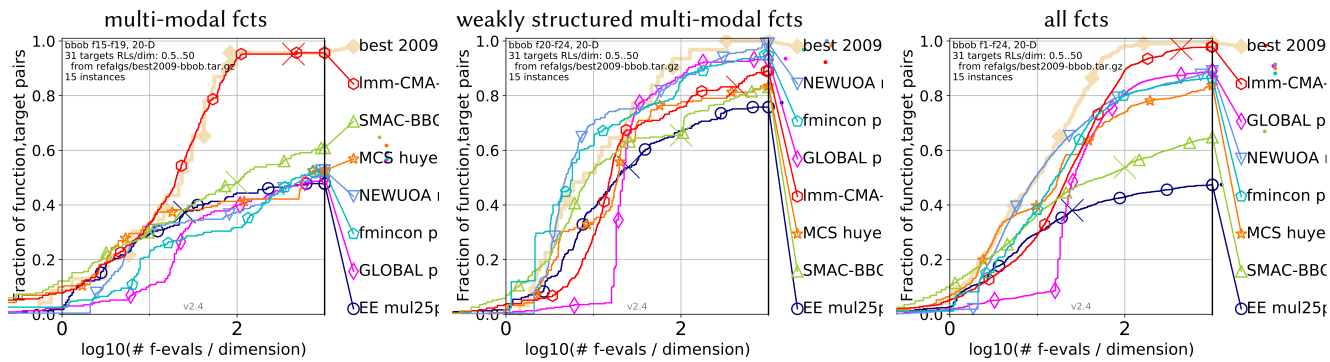

Figure 3: As in Figure 2, but for 20-D.

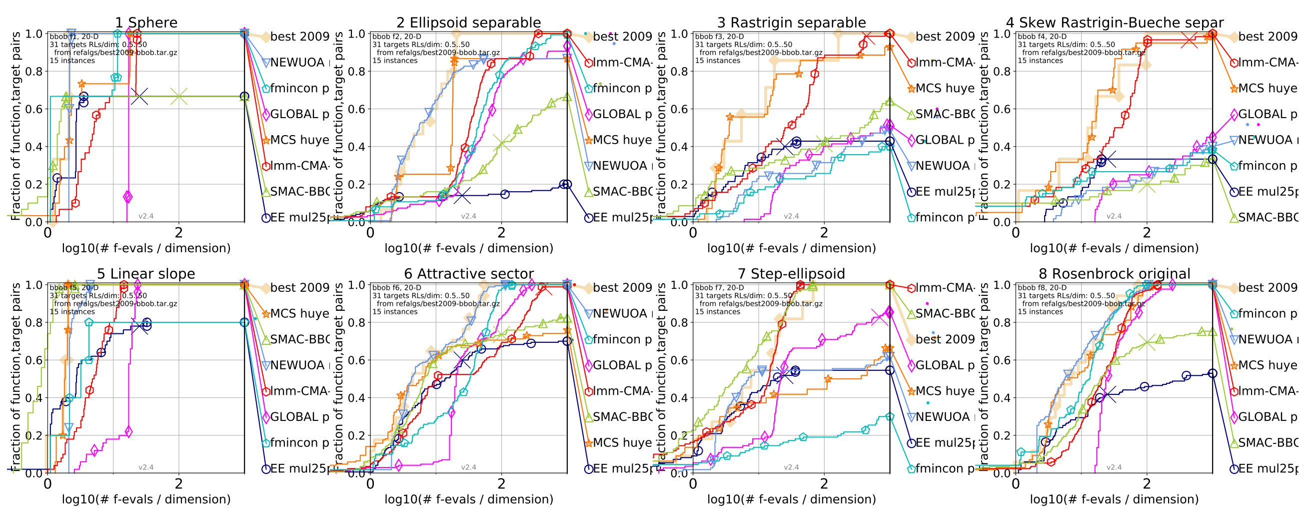

Figure 4: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of objective function evaluations divided by dimension (FEvals/DIM) in dimension 20 and for those targets in $10^{[-8..2]}$ that have just not been reached by the best algorithm from BBOB 2009 in a given budget of $k \times$ DIM, with 31 different values of k chosen equidistant in logscale within the interval $\{0.5, \ldots, 50\}$. Here we show just $f_1, \ldots, f_8$: functions $f_9, \ldots, f_{16}$ and $f_{17}, \ldots, f_{24}$ are shown in successive figures.

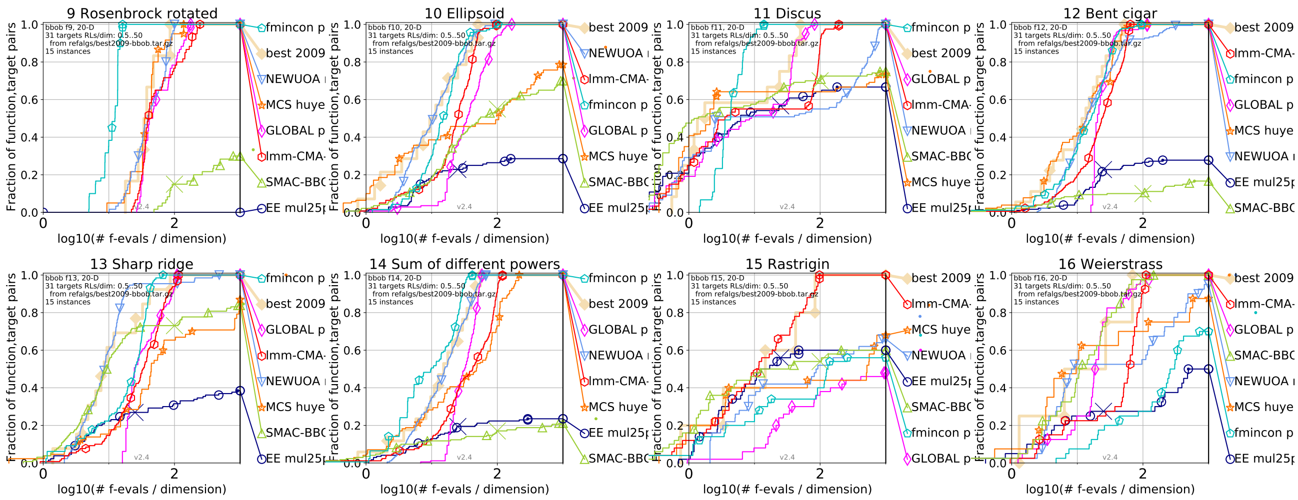

Figure 5: As in Figure 4, but for $f_9, \ldots, f_{16}$.

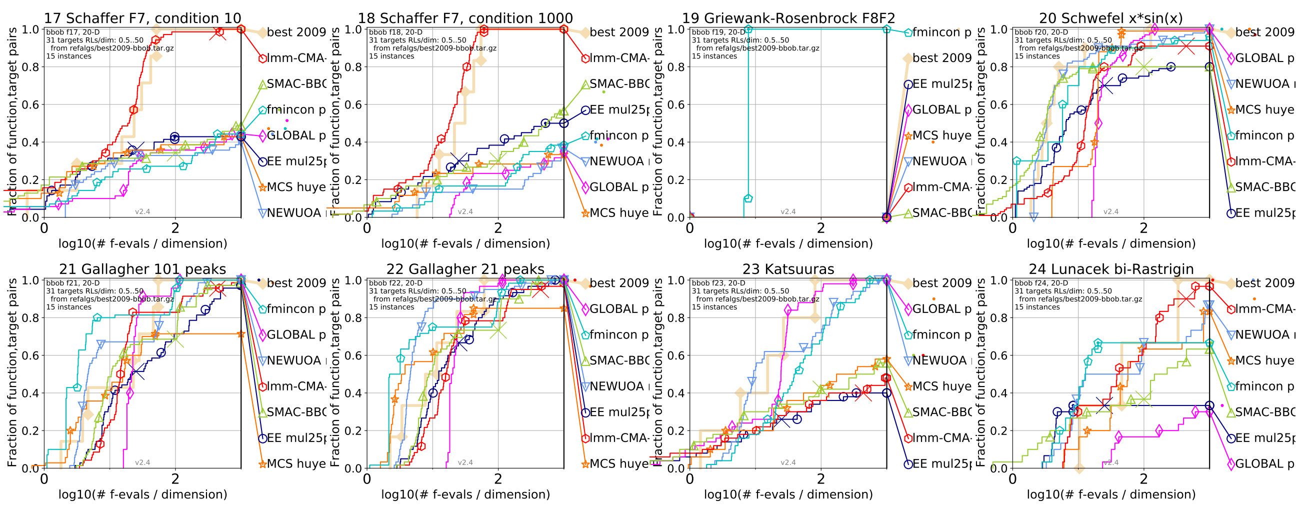

Figure 6: As in Figure 4, but for $f_{17}, \ldots, f_{24}$.

6 REMARKS

Some of the most important optimization problems in science and industry involve expensive objectives, e.g., the results of simulations. This fact and the experience of the present paper suggest introducing additional functionality in COCO to allow more precise benchmarking for extremely low function evaluation budgets.

Experiments on EXPLO2 along similar lines to those presented here indicate that using generic CMA-ES as the inner optimizer incurs a collateral runtime cost orders of magnitude beyond that of fmincon even for $D = 2$, which precludes careful benchmarking of that alternative.

Finally, though the runtime overhead of EXPLO2 scales favorably with dimension, it is still high for any given function evaluation: the surrogate is complicated and even a large-scale inner optimizer takes resources. EXPLO2 is therefore only suited for high-dimensional functions that are very expensive to evaluate, and it will generally be advisable to evaluate the suitability of *-CMA-ES as well in any particular application.

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*. 373–380.

[2] Martin D Buhmann. 2003. *Radial basis functions: theory and implementations*. Vol. 12. Cambridge.

[3] Carola Doerr, Hao Wang, Furong Ye, Sander van Rijn, and Thomas Bäck. 2018. IOHprofiler: A benchmarking and profiling tool for iterative optimization heuristics. *arXiv preprint arXiv:1810.05281* (2018).

[4] Ouassim Elhara, Konstantinos Varelas, Duc Nguyen, Tea Tusar, Dimo Brockhoff, Nikolaus Hansen, and Anne Auger. 2019. COCO: the large scale black-box optimization benchmarking (BBOB-largescale) test suite. *arXiv preprint arXiv:1903.06396* (2019).

#FEs/D	0.5	1.2	3	10	50	#succ
f1	*6.3e+1*:24	*4.0e+1*:42	*1.0e-8*:43	*1.0e-8*:43	*1.0e-8*:43	15/15
EE	1.5(0.6)	1.2(0.4)	∞	∞	∞ *501*	0/15
GLOBA	14(0.4)	8.1(0)	8.0(0)	8.0(0)	8.0(0)	15/15
MCS h	1.0(0.3)	1.00(0.1)	7.0(3)	7.0(3)	7.0(3)	15/15
NEWUO	1.8(0)	1.0(0.0)	**1.0**(0.0)$^{\star 4}$	**1.0**(0.0)$^{\star 4}$	**1.0**(0.0)$^{\star 4}$	15/15
lmm-C	2.5(2)	2.5(1)	10(0.2)	10(0.2)	10(0.2)	15/15
SMAC-	**0.81**(0.4)	0.67(0.2)	∞	∞	∞ *2000*	0/15
fminc	0.92(0)	**0.52**(0)*	5.1(0.4)	5.1(0.4)	5.1(0.5)	15/15
f2	*4.0e+6*:29	*2.5e+6*:42	*1.0e+5*:65	*1.0e+4*:207	*1.0e-8*:412	15/15
EE	0.57(0.5)	0.83(0.7)	∞	∞	∞ *501*	0/15
GLOBA	1.6(2)	2.8(3)	9.2(2)	7.2(1)	63(50)	13/15
MCS h	5.5(5)	4.0(4)	3.5(2)	1.4(0.8)	∞ *8e4*	0/15
NEWUO	1.4(0.3)	1(0)	**1.00**(0.4)	**1.00**(0.4)	303(130)	15/15
lmm-C	**0.52**(0.5)	**0.68**(0.7)	7.2(2)	3.8(1.0)	**14**(2)	15/15
SMAC-	0.53(0.4)	0.70(0.6)	23(18)	142(316)	∞ *2000*	0/15
fminc	2.6(0.4)	2.3(4)	10(9)	6.1(4)	22(10)	15/15
f3	*6.3e+2*:33	*4.0e+2*:44	*1.6e+2*:109	*1.0e+2*:255	*2.5e+1*:3277	15/15
EE	0.72(0.2)	1.8(0.7)	∞	∞	∞ *501*	0/15
GLOBA	7.8(4)	18(32)	441(739)	1438(930)	∞ *5e4*	0/15
MCS h	1(0.2)	**1.00**(0.1)	**1.0**(1.0)$^{\star 4}$	**1.00**(0.5)$^{\star 4}$	4.2(3)	15/15
NEWUO	2.5(1)	24(30)	360(595)	7456(7708)	∞ *1e5*	0/15
lmm-C	1.0(0.9)	2.3(1.0)	6.0(2)	4.8(0.7)	**2.1**(2)	14/15
SMAC-	**0.49**(0.5)	2.1(3)	125(211)	114(133)	∞ *2000*	0/15
fminc	6.6(13)	31(34)	6190(4670)	2.3e4(2e4)	∞ *4e5*	0/15
f4	*6.3e+2*:22	*4.0e+2*:91	*2.5e+2*:250	*1.6e+2*:332	*6.3e+1*:1927	15/15
EE	5.3(4)	4.0(2)	∞	∞	∞ *501*	0/15
GLOBA	17(4)	78(173)	186(239)	3081(2405)	∞ *8e4*	0/15
MCS h	**0.98**(0.9)	**1**(0.6)*	**1.0**(0.2)*	**1.00**(0.1)$^{\star 4}$	4.1(5)	15/15
NEWUO	37(79)	48(96)	104(106)	1703(1866)	∞ *2e5*	0/15
lmm-C	1.2(1)	2.2(0.9)	2.0(0.8)	3.0(0.7)	**2.0**(2)	13/15
SMAC-	6.7(9)	102(71)	∞	∞	∞ *2000*	0/15
fminc	1.3(2)	42(80)	525(984)	5108(5132)	∞ *4e5*	0/15
f5	*2.5e+2*:19	*1.6e+2*:34	*1.0e-8*:41	*1.0e-8*:41	*1.0e-8*:41	15/15
EE	1.1(0.3)	0.84(0.4)	∞	∞	∞ *501*	0/15
GLOBA	5.6(6)	10(0.2)	11(0.9)	11(0.9)	11(0.9)	15/15
MCS h	1.4(0.0)	1.0(7e-3)	0.99(0.0)	0.99(0.0)	0.99(0.0)	15/15
NEWUO	2.2(0)	1.3(0.0)	1.6(0.5)	1.6(0.4)	1.6(0.5)	15/15
lmm-C	1.8(0.6)	2.1(0.5)	6.1(1)	6.1(0.9)	6.1(0.9)	15/15
SMAC-	**0.46**(0.1)$^{\star 2}$	**0.33**(0.1)$^{\star 4}$	**0.66**(0.3)$^{\star 2}$	**0.66**(0.3)$^{\star 2}$	**0.66**(0.3)$^{\star 2}$	15/15
fminc	1.2(0)	1.3(0)	3.0e4(4e4)	3.0e4(3e4)	3.0e4(8e4)	4/15
f6	*2.5e+5*:16	*6.3e+4*:43	*1.6e+4*:62	*1.6e+2*:353	*1.6e+1*:1078	15/15
EE	1.7(1)	1.3(0.7)	1.9(1)	10(9)	∞ *501*	0/15
GLOBA	11(10)	7.9(0.7)	6.1(0.9)	3.0(2)	3.5(1)	15/15
MCS h	**1**(1)	**1.0**(0.5)	1.3(1)	6.9(12)	1102(749)	1/15
NEWUO	2.4(1)	1.3(0.3)	**1.00**(0.3)	**1.0**(0.7)	**1.0**(0.2)	15/15
lmm-C	1.6(1.0)	1.6(0.8)	2.0(0.7)	6.8(4)	9.0(5)	11/15
SMAC-	1.6(1)	1.2(0.9)	1.6(1)	2.8(3)	∞ *2000*	0/15
fminc	6.0(15)	5.3(8)	5.9(5)	2.6(1)	1.6(0.7)	15/15
f7	*1.0e+3*:11	*4.0e+2*:39	*2.5e+2*:74	*6.3e+1*:319	*1.0e+1*:1351	15/15
EE	1.2(1)	1.2(0.6)	1.2(0.5)	∞	∞ *501*	0/15
GLOBA	4.4(7)	7.4(3)	4.9(0.4)	4.3(6)	∞ *1e4*	0/15
MCS h	0.96(0.8)	1(0.5)	3.7(4)	66(86)	∞ *8e4*	0/15
NEWUO	3.5(0.2)	1.4(0.3)	1.0(0.6)	34(69)	∞ *5e5*	0/15
lmm-C	**0.47**(0.6)	1.2(1)	1.7(1)	1.0(0.2)	**0.48**(0.1)	15/15
SMAC-	0.56(0.7)	**0.61**(0.5)	**0.49**(0.3)	**0.39**(0.3)*	0.57(0.3)	15/15
fminc	2.8(3)	166(161)	468(390)	∞	∞ *2e5*	0/15
f8	*4.0e+4*:19	*2.5e+4*:35	*4.0e+3*:67	*2.5e+2*:231	*1.6e+1*:1470	15/15
EE	1.9(0.9)	1.3(0.5)	2.2(2)	∞	∞ *501*	0/15
GLOBA	18(0.6)	10(0.5)	5.7(0.3)	3.0(2)	1.6(0.5)	15/15
MCS h	**0.99**(0.9)	1.0(0.6)	1.3(0.9)	1.3(2)	1.2(0.3)	15/15
NEWUO	2.2(0)	1.3(0.4)	**1.0**(0.4)	**1.00**(0.9)	1.0(0.4)	15/15
lmm-C	2.1(3)	2.6(1)	3.3(1)	2.5(0.6)	1.3(0.4)	15/15
SMAC-	1.4(2)	1.5(1)	2.5(0.9)	4.1(2)	∞ *2000*	0/15
fminc	1.1(1)	**0.99**(0.9)	1.6(1)	1.4(0.5)	**0.84**(0.2)	15/15
f9	*1.0e+2*:357	*6.3e+1*:560	*4.0e+1*:684	*2.5e+1*:756	*1.0e+1*:1716	15/15
EE	∞	∞	∞	∞	∞ *501*	0/15
GLOBA	2.0(1)	1.5(0.9)	1.3(0.5)	1.3(0.5)	1.7(0.3)	15/15
MCS h	1.00(0.3)	1.0(0.2)	1.1(0.2)	1.1(0.1)	1.00(0.5)	15/15
NEWUO	1.3(0.6)	1.1(0.8)	1.00(0.7)	1.0(0.4)	1.0(0.2)	15/15
lmm-C	1.6(0.2)	1.3(0.2)	1.1(0.3)	1.2(0.1)	2.1(0.5)	15/15
SMAC-	5.0(3)	26(17)	∞	∞	∞ *2000*	0/15
fminc	**0.30**(0.1)$^{\star 4}$	**0.31**(2e-3)$^{\star 4}$	**0.32**(0.0)$^{\star 4}$	**0.34**(0.0)$^{\star 4}$	**0.17**(0.0)$^{\star 4}$	15/15
f10	*1.6e+6*:15	*1.0e+6*:27	*4.0e+5*:70	*6.3e+4*:231	*4.0e+3*:1015	15/15
EE	5.3(4)	4.1(2)	22(24)	∞	∞ *501*	0/15
GLOBA	14(9)	13(3)	6.0(0.8)	3.3(1)	2.1(0.7)	15/15
MCS h	**1**(0.3)*	**1.0**(0.6)$^{\star 2}$	**1.0**(0.5)	14(19)	1162(1074)	1/15
NEWUO	4.2(1)	2.6(1)	1.4(0.6)	**1.00**(0.4)	1(0.6)	15/15
lmm-C	3.7(4)	3.6(2)	4.2(1)	2.3(0.6)	1.00(0.1)	15/15
SMAC-	3.7(5)	5.6(4)	6.2(5)	18(18)	∞ *2000*	0/15
fminc	5.2(3)	3.9(2)	2.7(0.8)	1.7(0.8)	**0.67**(0.5)	15/15

#FEs/D	0.5	1.2	3	10	50	#succ
f11	*4.0e+4*:11	*2.5e+3*:27	*1.6e+2*:313	*1.0e+2*:481	*1.0e+1*:1002	15/15
EE	1.3(1)	4.3(8)	6.7(9)	∞	∞ *501*	0/15
GLOBA	1.7(1)	3.0(4)	2.1(0.6)	1.5(0.2)	1.2(0.4)	15/15
MCS h	1.3(0.6)	1(0.6)	4.8(6)	53(68)	∞ *8e4*	0/15
NEWUO	2.1(2)	1.4(0.8)	16(7)	15(4)	15(3)	15/15
lmm-C	1.3(1)	1.4(0.9)	4.5(3)	3.7(1)	2.1(0.2)	15/15
SMAC-	**0.60**(0.3)	**0.68**(0.5)	2.4(3)	7.3(16)	∞ *2000*	0/15
fminc	4.6(2)	2.7(1)	**0.33**(0.2)	**0.22**(0.1)$^{\star 2}$	**0.16**(0.0)$^{\star 4}$	15/15
f12	*1.0e+8*:23	*6.3e+7*:39	*2.5e+7*:76	*4.0e+6*:209	*1.0e+1*:1042	15/15
EE	5.1(4)	5.7(2)	8.1(9)	∞	∞ *501*	0/15
GLOBA	14(3)	8.8(0.5)	4.8(0.1)	1.9(0.2)	1.00(0.9)	15/15
MCS h	**1.0**(0.9)	**1.0**(0.5)	**1.00**(0.3)*	**1.00**(0.5)	1.1(0.8)	15/15
NEWUO	3.5(1)	2.5(0.8)	1.9(0.9)	1.2(0.5)	3.0(3)	15/15
lmm-C	2.0(1)	2.4(1)	2.6(0.6)	1.9(0.3)	1.1(0.1)	15/15
SMAC-	2.8(3)	38(51)	∞	∞	∞ *2000*	0/15
fminc	2.0(2)	1.8(0.9)	1.6(0.7)	1.4(0.7)	**0.81**(0.5)	15/15
f13	*1.6e+3*:28	*1.0e+3*:64	*6.3e+2*:79	*4.0e+1*:211	*2.5e+0*:1724	15/15
EE	1.5(0.6)	1.2(0.3)	1.8(0.6)	∞	∞ *501*	0/15
GLOBA	11(1)	5.7(0)	4.8(0.3)	3.4(0.7)	1.1(0.1)	15/15
MCS h	0.99(0.7)	1.3(0.8)	3.4(0.6)	48(54)	38(25)	12/15
NEWUO	1.7(0.3)	1.0(0.1)	0.99(0.2)	**1.00**(0.2)$^{\star 3}$	1.5(3)	15/15
lmm-C	1.6(0.9)	2.6(1)	3.3(0.8)	4.2(0.5)	1.1(0.2)	15/15
SMAC-	**0.80**(0.4)	**0.66**(0.2)	**0.84**(0.1)*	1.4(0.2)	∞ *2000*	0/15
fminc	1.1(0.4)	1.2(0.9)	1.8(1)	2.8(0.5)	**0.51**(0.1)	15/15
f14	*2.5e+1*:15	*1.6e+1*:42	*1.0e+1*:75	*1.6e+0*:219	*6.3e-4*:1106	15/15
EE	2.9(2)	3.4(3)	3.7(3)	∞	∞ *501*	0/15
GLOBA	19(8)	8.3(0.5)	5.0(0.3)	2.3(0.3)	1(0.1)	15/15
MCS h	**0.97**(0.5)	**1.0**(0.4)	1(0.5)	1.9(0.9)	3.3(1)	15/15
NEWUO	4.2(1)	2.1(1)	1.5(0.8)	1.00(0.5)	1.0(0.2)	15/15
lmm-C	3.7(5)	2.9(2)	3.0(1)	2.2(0.4)	1.9(0.1)	15/15
SMAC-	2.0(3)	3.4(7)	19(11)	∞	∞ *2000*	0/15
fminc	1.9(0.7)	1.1(0.5)	**0.74**(0.1)	**0.49**(0.1)$^{\star 3}$	**0.60**(0.1)	15/15
f15	*6.3e+2*:15	*4.0e+2*:67	*2.5e+2*:292	*1.6e+2*:846	*1.0e+2*:1671	15/15
EE	2.0(1)	1.2(1.0)	1.4(0.7)	∞	∞ *501*	0/15
GLOBA	15(11)	24(26)	110(97)	∞	∞ *2e4*	0/15
MCS h	**1.0**(0.4)	**1.0**(0.3)	35(23)	96(78)	70(72)	7/15
NEWUO	4.4(3)	11(17)	14(18)	41(32)	1078(1235)	1/15
lmm-C	1.4(2)	1.3(0.7)	**0.77**(0.4)	**0.72**(0.3)	**0.86**(0.3)	15/15
SMAC-	1.2(2)	2.9(0.9)	2.8(5)	∞	∞ *2000*	0/15
fminc	30(72)	20(19)	42(65)	323(287)	1681(1968)	2/15
f16	*4.0e+1*:26	*2.5e+1*:127	*1.6e+1*:540	*1.6e+1*:540	*1.0e+1*:1384	15/15
EE	2.8(2)	27(38)	∞	∞	∞ *501*	0/15
GLOBA	4.5(4)	2.7(0.3)	1.00(0.7)	1.00(0.7)	1.00(1)	15/15
MCS h	**1.3**(1)	17(40)	4.4(10)	4.4(5)	11(13)	15/15
NEWUO	2.5(0.8)	6.5(9)	7.8(11)	7.8(6)	16(23)	15/15
lmm-C	4.9(8)	8.1(6)	2.7(0.8)	2.7(0.8)	1.2(0.4)	15/15
SMAC-	2.4(3)	**1.5**(0.9)	**0.78**(0.3)	**0.78**(0.5)	**0.76**(0.4)	14/15
fminc	22(30)	29(23)	51(87)	51(35)	675(1769)	5/15
f17	*1.6e+1*:11	*1.0e+1*:63	*6.3e+0*:305	*4.0e+0*:468	*1.0e+0*:1030	15/15
EE	1.8(0.6)	1.3(0.6)	2.1(2)	∞	∞ *501*	0/15
GLOBA	9.0(13)	6.2(2)	33(77)	582(531)	∞ *7e4*	0/15
MCS h	0.96(2)	1(0.7)	8.5(18)	587(605)	∞ *8e4*	0/15
NEWUO	5.1(3)	16(4)	55(131)	3447(5522)	∞ *2e6*	0/15
lmm-C	0.59(0)	1(0.9)	**0.65**(0.3)	**0.79**(0.2)$^{\star 3}$	**1.4**(0.4)	14/15
SMAC-	**0.50**(0.9)	**0.92**(1.0)	15(13)	61(89)	∞ *2000*	0/15
fminc	20(65)	21(28)	14(14)	728(495)	∞ *4e5*	0/15
f18	*4.0e+1*:116	*2.5e+1*:252	*1.6e+1*:430	*1.0e+1*:621	*4.0e+0*:1090	15/15
EE	0.59(0.9)	1.8(1)	8.7(13)	∞	∞ *501*	0/15
GLOBA	11(25)	56(98)	428(815)	∞	∞ *7e4*	0/15
MCS h	1.8(2)	10(12)	491(725)	∞	∞ *8e4*	0/15
NEWUO	20(43)	47(91)	1013(1095)	1.2e4(1e4)	∞ *2e6*	0/15
lmm-C	0.41(0.7)	**0.57**(0.3)	**0.77**(0.3)$^{\star 3}$	**0.76**(0.3)$^{\star 3}$	**0.85**(0.3)	15/15
SMAC-	**0.31**(0.4)	8.3(20)	20(15)	22(15)	∞ *2000*	0/15
fminc	7.0(6)	18(12)	823(1279)	∞	∞ *4e5*	0/15
f19	*1.6e-1*:2.5e5	*1.0e-1*:3.4e5	*6.3e-2*:3.4e5	*4.0e-2*:3.4e5	*2.5e-2*:3.4e5	3/15
EE	∞	∞	∞	∞	∞ *501*	0/15
GLOBA	∞	∞	∞	∞	∞ *5e4*	0/15
MCS h	1.4(1)	1(0.7)	1(0.6)	1(2)	1(2)	3/15
NEWUO	∞	∞	∞	∞	∞ *2e6*	0/15
lmm-C	∞	∞	∞	∞	∞ *8805*	0/15
SMAC-	∞	∞	∞	∞	∞ *2000*	0/15
fminc	**6.2e-4**(2e-5)$^{\star 4}$	**7.3e-4**(7e-7)$^{\star 4}$	**8.8e-4**(7e-7)$^{\star 4}$	**1.0e-3**(9e-5)$^{\star 4}$	**1.2e-3**(9e-5)$^{\star 4}$	15/15
f20	*1.6e+4*:38	*1.0e+4*:42	*2.5e+2*:62	*2.5e+0*:250	*1.6e+0*:2536	15/15
EE	0.68(0.1)	0.87(0.4)	2.5(2)	∞	∞ *501*	0/15
GLOBA	8.7(0.3)	8.0(0.2)	6.2(0.3)	3.8(1)	**1.5**(2)	15/15
MCS h	2.1(0)	3.2(3)	5.3(3)	1.8(0.4)	4.7(11)	14/15
NEWUO	1.1(0)	1.0(0)	1.0(0.2)	**1.0**(0.5)$^{\star 2}$	2.1(2)	15/15
lmm-C	0.92(1.0)	1.4(2)	3.9(0.9)	6.6(0.8)	23(37)	2/15
SMAC-	**0.25**(0.1)*	**0.46**(0.2)	**0.91**(0.2)	∞	∞ *2000*	0/15
fminc	0.61(0)	0.55(0)	1.5(0)	2.2(0.5)	6.9(7)	15/15

Table 1: Expected runtime (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 in dimension 20. This ERT ratio and, in braces as dispersion measure, the half difference between 10 and 90%-tile of bootstrapped run lengths appear for each algorithm and run-length based target, the corresponding reference ERT (preceded by the target Δf-value in *italics*) in the first row. #succ is the number of trials that reached the target value of the last column. 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 by the number of functions (24). A ↓ indicates the same tested against the best algorithm from BBOB 2009. Best results are printed in bold.

Data produced with COCO v2.4
produced with COCO v2.4

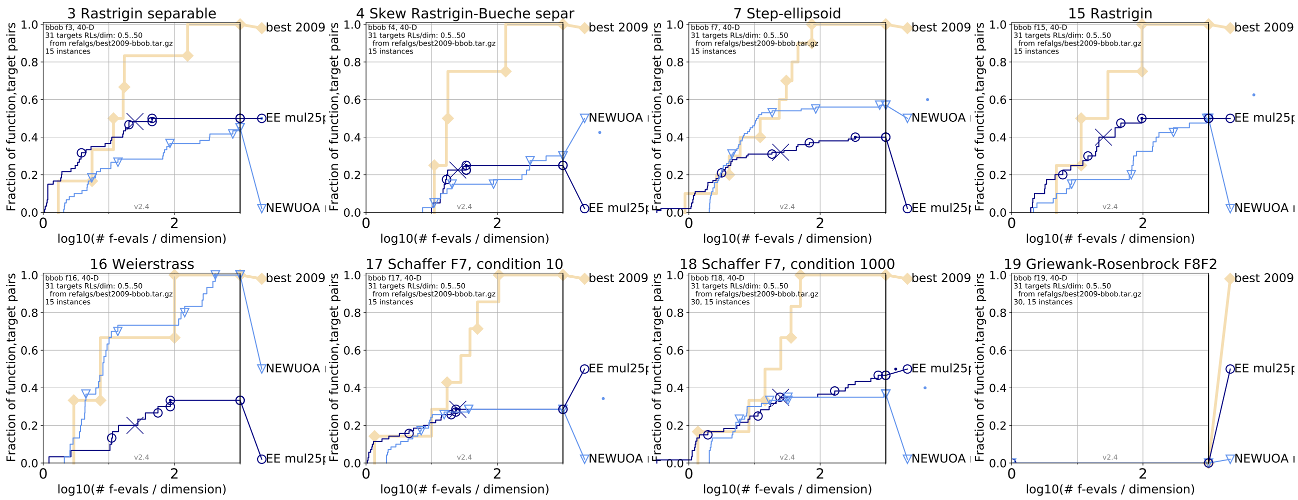

Figure 7: As in Figure 4, but for dimension 40 and $f_3, f_4, f_7, f_{15}, \ldots, f_{19}$.

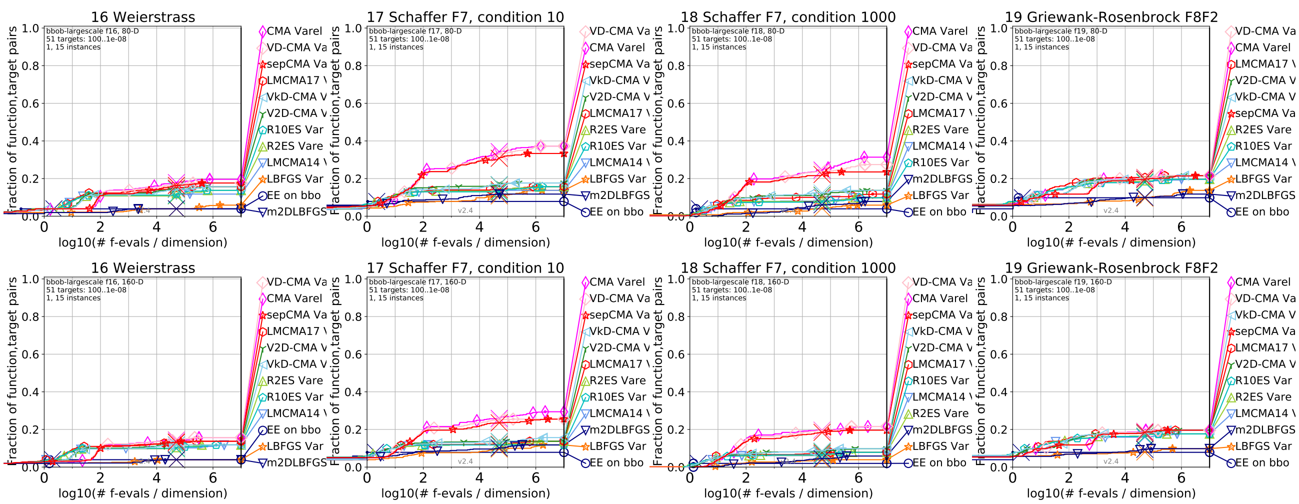

Figure 8: As in Figure 4, but for dimension 80 (top row) and 160 (bottom row) and $f_{16}, \ldots, f_{19}$ on the `bbob-largescale` testbed. The Rastrigin function f_{15} has degenerate results for all algorithms and so is not shown. Note that [12] points out that algorithms are only comparable up to the smallest budget given to any of them, corresponding in our case exactly to the location of crosses (×), which indicate where bootstrapping of experimental data begins to estimate results for larger numbers of function evaluations. These locations are very far to the left, though EXPL02 (here indicated as EE) is consistently a top performer there.

#FEs/D	0.5	1.2	3	10	50	#succ
f21	*6.3e+1*:36	*4.0e+1*:77	*4.0e+1*:77	*1.6e+1*:456	*4.0e+0*:1094	15/15
EE	3.5(2)	2.9(3)	2.9(2)	1.3(1.0)	6.6(12)	1/15
GLOBA	9.2(0.7)	5.2(0.8)	5.2(0.8)	1.0(0.2)	1.00(0.9)	15/15
MCS h	0.99(0.5)	1.0(0.7)	1.0(0.7)	31(44)	36(39)	14/15
NEWUO	1.8(0.2)	0.99(0.2)	0.99(0.2)	1.4(3)	2.9(2)	15/15
lmm-C	3.5(1)	2.5(1)	2.5(1)	1.3(0.1)	4.3(9)	11/15
SMAC-	7.5(4)	4.2(7)	4.2(8)	2.7(4)	5.2(3)	4/15
fminc	**0.81**(0.3)	**0.64**(0.2)	**0.64**(0.3)	**0.96**(0.9)	**0.92**(1)	15/15
f22	*6.3e+1*:45	*4.0e+1*:68	*4.0e+1*:68	*1.6e+1*:231	*6.3e+0*:1219	15/15
EE	2.4(0.7)	2.8(3)	2.8(3)	2.6(3)	2.9(3)	2/15
GLOBA	7.9(0.6)	5.8(0.9)	5.8(0.9)	2.0(0.9)	**1.0**(1)	15/15
MCS h	1(0.3)	**0.99**(0.4)	**0.99**(0.3)	33(76)	20(27)	13/15
NEWUO	1.5(0.2)	1.2(0.3)	1.2(0.4)	**1.00**(1)	1.4(3)	15/15
lmm-C	3.1(2)	5.8(2)	5.8(11)	5.9(13)	3.8(8)	11/15
SMAC-	10(13)	7.2(22)	7.2(9)	4.2(7)	2.0(2)	7/15
fminc	**0.99**(0.4)	2.5(0.2)	2.5(7)	2.9(4)	3.5(9)	15/15

#FEs/D	0.5	1.2	3	10	50	#succ
f23	*6.3e+0*:29	*4.0e+0*:118	*2.5e+0*:306	*2.5e+0*:306	*1.0e+0*:1614	15/15
EE	**1.5**(2)	11(14)	∞	∞	∞ *501*	0/15
GLOBA	1.8(2)	2.9(1)	**1.7**(0.2)	**1.7**(0.2)	**1.00**(0.7)	15/15
MCS h	1.5(0.9)	18(26)	29(31)	29(31)	124(103)	5/15
NEWUO	7.2(19)	**2.8**(6)	2.1(3)	2.1(3)	3.5(4)	15/15
lmm-C	1.9(3)	8.2(5)	408(548)	408(296)	∞ *8823*	0/15
SMAC-	1.5(2)	5.0(4)	46(68)	46(75)	∞ *2000*	0/15
fminc	4.9(5)	4.8(7)	3.2(2)	3.2(2)	4.4(4)	15/15
f24	*2.5e+2*:208	*1.6e+2*:918	*1.0e+2*:6628	*6.3e+1*:9885	*4.0e+1*:31629	15/15
EE	**0.40**(0.2)	∞	∞	∞	∞ *501*	0/15
GLOBA	31(95)	∞	∞	∞	∞ *3e4*	0/15
MCS h	1.8(1)	2.7(4)	12(33)	∞	∞ *8e4*	0/15
NEWUO	1.1(2)	1.2(2)	4.3(5)	247(334)	∞ *2e5*	0/15
lmm-C	0.74(0.2)	**1.1**(0.5)	**1.4**(0.8)	**1.2**(0.9)	**1.2**(1)	3/15
SMAC-	0.65(1)	10(5)	∞	∞	∞ *2000*	0/15
fminc	0.51(0.3)	12(1)	886(1345)	∞	∞ *4e5*	0/15

Table 2: Continuation of Table 1.

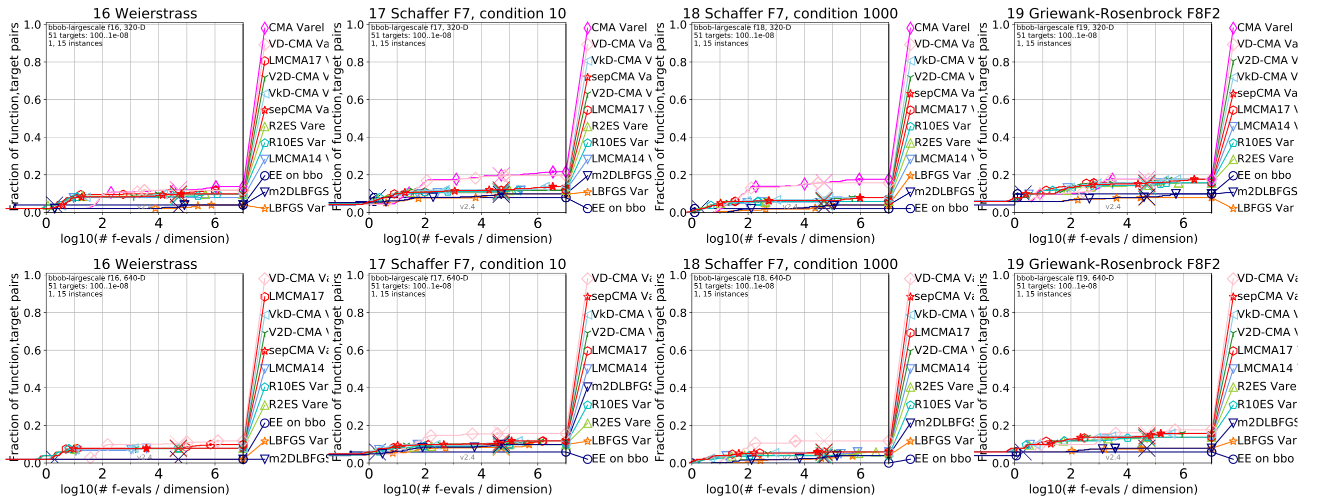

Figure 9: As in Figure 8, but for dimensions 320 (top row) and 640 (bottom row).

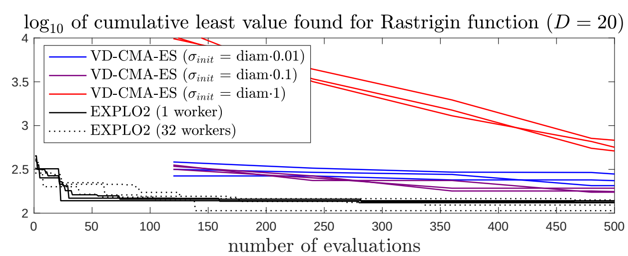

Figure 10: A comparison of VD-CMA-ES and EXPLO2 on the Rastrigin function (without rotations or shifts, but otherwise corresponding to f_{15} in the BBOB testbed) on $[-5.12, 5.12]^D$ for $D = 20$ and a budget of 500 function evaluations. For VD-CMA-ES, various initial step sizes σ_{init} were selected as shown and logging is intermittent; meanwhile, for EXPLO2, we considered $n_\parallel \in \{1, 32\}$ parallel workers. Each algorithm/parameter pair was run three times, all shown. Note that the initialization phase of VD-CMA-ES is not plotted.

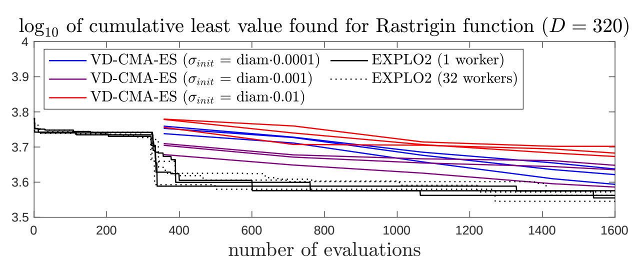

Figure 11: As in Figure 10, but for $D = 320$ and a budget of 1600.

[5] 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.

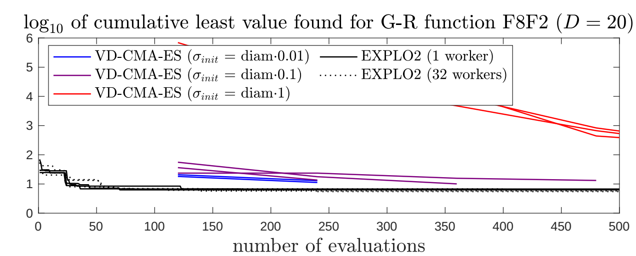

Figure 12: As in Figure 10, but for the Griewank-Rosenbrock function F8F2 (without rotations or shifts, but otherwise corresponding to f_{19} in the BBOB testbed) on $[-5, 5]^D$ for $D = 20$. Note that VD-CMA-ES often terminates before the budget is reached.

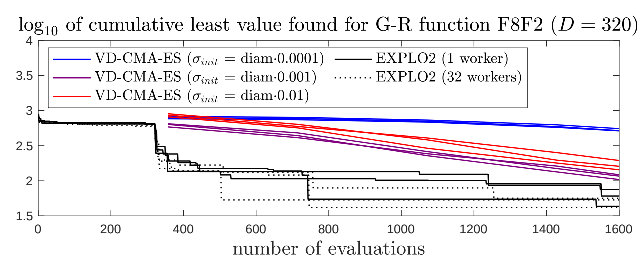

Figure 13: As in Figure 12, but for $D = 320$ and a budget of 1600.

[6] Guohua Gao, Jeroen C Vink, Chaohui Chen, Yaakoub El Khamra, and Mohammadali Tarrahi. 2017. Distributed Gauss-Newton optimization method for history matching problems with multiple best matches. *Computational Geosciences* 21, 5 (2017), 1325–1342.

[7] Raphael T Haftka, Diane Villanueva, and Anirban Chaudhuri. 2016. Parallel surrogate-assisted global optimization with expensive functions–a survey. *Structural and Multidisciplinary Optimization* 54, 1 (2016), 3–13.

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

[9] Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. 2021. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software* 36, 1 (2021), 114–144.

[10] 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.

[11] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation* 11, 1 (2003), 1–18.

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

[13] Steve Huntsman. 2022. Parallel black-box optimization of expensive high-dimensional multimodal functions via magnitude. *arXiv preprint arXiv:2201.11677* (2022).

[14] Najeeb Khan. 2018. A parallel implementation of the covariance matrix adaptation evolution strategy. *arXiv preprint arXiv:1805.11201* (2018).

[15] Tom Leinster. 2021. *Entropy and Diversity: the Axiomatic Approach*. Cambridge.

[16] Tom Leinster and Mark W Meckes. [n.d.]. The magnitude of a metric space: from category theory to geometric measure theory. In *Measure Theory in Non-Smooth Spaces*, Nicola Gigli (Ed.).

[17] Kenneth Price. 1997. Differential evolution vs. the functions of the second ICEO. In *Proceedings of the IEEE International Congress on Evolutionary Computation*. IEEE, Piscataway, NJ, USA, 153–157. https://doi.org/10.1109/ICEC.1997.592287

[18] Frederik Rehbach, Martin Zaefferer, Jörg Stork, and Thomas Bartz-Beielstein. 2018. Comparison of parallel surrogate-assisted optimization approaches. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 1348–1355.

[19] Raymond Ros. 2009. Benchmarking the NEWUOA on the BBOB-2009 function testbed. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. 2421–2428.

[20] Ingo Steinwart and Andreas Christmann. 2008. *Support vector machines*. Springer.

[21] Tea Tušar, Dimo Brockhoff, and Nikolaus Hansen. 2019. Mixed-integer benchmark problems for single-and bi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 718–726.

[22] Tea Tušar, Nikolaus Hansen, and Dimo Brockhoff. 2017. Anytime Benchmarking of Budget-Dependent Algorithms with the COCO Platform. In *IS 2017-International multiconference Information Society*. 1–4.

[23] Konstantinos Varelas. 2019. Benchmarking large scale variants of cma-es and l-bfgs-b on the bbob-largescale testbed. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1937–1945.

[24] Konstantinos Varelas, Anne Auger, Dimo Brockhoff, Nikolaus Hansen, Ouassim Ait ElHara, Yann Semet, Rami Kassab, and Frédéric Barbaresco. 2018. A comparative study of large-scale variants of CMA-ES. In *International Conference on Parallel Problem Solving from Nature*. Springer, 3–15.

[25] Wei Xia and Christine Shoemaker. 2021. GOPS: efficient RBF surrogate global optimization algorithm with high dimensions and many parallel processors including application to multimodal water quality PDE model calibration. *Optimization and Engineering* 22, 4 (2021), 2741–2777.