



HAL
open science

Multiobjective optimization with a quadratic surrogate-assisted CMA-ES

Mohamed Gharafi, Nikolaus Hansen, Dimo Brockhoff, Rodolphe Le Riche

► **To cite this version:**

Mohamed Gharafi, Nikolaus Hansen, Dimo Brockhoff, Rodolphe Le Riche. Multiobjective optimization with a quadratic surrogate-assisted CMA-ES. GECCO 2023 - Genetic and Evolutionary Computation Conference, GECCO, Jul 2023, Lisbon, Portugal. 10.1145/3583131.3590492 . hal-04078483

HAL Id: hal-04078483

<https://hal.science/hal-04078483v1>

Submitted on 23 Apr 2023

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.



Distributed under a Creative Commons Attribution 4.0 International License

Multiobjective optimization with a quadratic surrogate-assisted CMA-ES

Mohamed Gharafi

mohamed.gharafi@inria.fr

Inria, CMAP, Ecole Polytechnique, IP Paris
Paris, France

Dimo Brockhoff

forename.lastname@inria.fr

Inria, CMAP, Ecole Polytechnique, IP Paris
Paris, France

Nikolaus Hansen

forename.lastname@inria.fr

Inria, CMAP, Ecole Polytechnique, IP Paris
Paris, France

Rodolphe Le Riche

leriche@emse.fr

CNRS LIMOS
St-Etienne, France

ABSTRACT

We present a surrogate-assisted multiobjective optimization algorithm. The aggregation of the objectives relies on the Uncrowded Hypervolume Improvement (UHVI) which is partly replaced by a linear-quadratic surrogate that is integrated into the CMA-ES algorithm. Surrogating the UHVI poses two challenges. First, the UHVI is a dynamic function, changing with the empirical Pareto set. Second, it is a composite function, defined differently for dominated and nondominated points. The presented algorithm is thought to be used with expensive functions of moderate dimension (up to about 50) with a quadratic surrogate which is updated based on its ranking ability. We report numerical experiments which include tests on the COCO benchmark. The algorithm shows in particular linear convergence on the double sphere function with a convergence rate that is 6-20 times faster than without surrogate assistance.

CCS CONCEPTS

• **Mathematics of computing** → **Stochastic control and optimization**; • **Computing methodologies** → *Modeling methodologies; Model verification and validation.*

KEYWORDS

multiobjective optimization, CMA-ES, evolution strategies, surrogate-assisted optimization, quadratic metamodel

ACM Reference Format:

Mohamed Gharafi, Nikolaus Hansen, Dimo Brockhoff, and Rodolphe Le Riche. 2023. Multiobjective optimization with a quadratic surrogate-assisted CMA-ES. In *Genetic and Evolutionary Computation Conference (GECCO '23)*, July 15–19, 2023, Lisbon, Portugal. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3583131.3590492>

1 INTRODUCTION

Many practical optimization problems have competing or even incommensurable objectives, the computation of which takes substantial numerical effort. Multiobjective optimization (MO) algorithms

tackle such problems but they are typically more costly than single-objective algorithms as they approximate the Pareto set which may be infinite in size. In this context, MO algorithms may be equipped with surrogates of the objectives in order to circumvent evaluations of the expensive objectives and hence save computing time.

The use of surrogates (or metamodels) dates back to the origins of the single-objective optimization methods in the 50's when it was introduced as the response surface methodology [4]. The addition of surrogates to MO algorithms, while being much more recent ([16] is one of the early contributions), has already received much attention. Recent reviews can be found in [1, 7]. Including surrogates in MO algorithms raises four main questions.

Which quantity should be surrogated? In MO, one can choose to surrogate the objectives [19] or a function aggregating them, for example the hypervolume improvement. In [18], the aggregating function is learned through a support vector machine classifying Pareto and non-Pareto points. A taxonomy of the possible combinations of aggregated functions of the objectives and constraints which are then metamodelled is presented in [9].

With which surrogate? Many possible surrogates exist such as quadratic response surfaces, Gaussian processes, support-vector machines, and neural networks. A basic rationale for choosing the surrogate is to compromise between the numerical cost of building the surrogate (number of points needed and algorithmic complexity) and the required accuracy, which depends on the problem at hand. In particular, it should be decided whether the scope of the surrogate is local to a trusted region or global to the entire design space. The targeted problem dimension is relevant for the choice too.

How to handle simultaneously the primary goal of solving an optimization problem and the secondary goal of learning the surrogate? This question has given rise to the notion of acquisition criteria, popular in Bayesian optimization. In the MO context, points are commonly selected based on the expected hypervolume improvement [3].

How is the surrogate used in the optimization algorithm? A surrogate can replace calls to the objective functions during all or some phases of the algorithm (as in the “local searches” of [19]). It can also filter out some points predicted to be bad (e.g., in [18]).

The current article studies the addition of a linear-quadratic surrogate to a multiobjective version of CMA-ES. We surrogate the Uncrowded Hypervolume Improvement (UHVI) from COMO-CMA-ES [21]. The approach is quite straightforward as we can simply

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

GECCO '23, July 15–19, 2023, Lisbon, Portugal

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0119-1/23/07...\$15.00

<https://doi.org/10.1145/3583131.3590492>

replace the single-objective CMA-ES [13] in the COMO-CMA-ES with the surrogate based lq-CMA-ES [11].

Our idea is that the aggregated criterion is what really matters in MO and we expect to better control the effects of the surrogate errors by directly working with the UHVI figure of merit. This is a way to account for the coupling between the different objectives which independent surrogates cannot do. In passing, this allows to build only one surrogate (as opposed to one for each objective) which is an ingredient for numerical sparsity.

A linear-quadratic surrogate is conceptually simple and numerically stable and we target problem dimensions larger than ten for which it is not clear that more flexible surrogates can be learned [11]. The acquisition criterion is directly the value of the UHVI. Contrarily to Bayesian optimizers [20], there is no need for a specific acquisition criterion here because the surrogate is not interpolating and the diversity is controlled through the randomness of the evolutionary sampling.

Our surrogate replaces calls to the objective functions. As a first step towards rank-invariance, tests of its *ranking* ability control the proportion of points for which the true functions are calculated. This work and [17] share the use of quadratic surrogates but in [17] the objectives are approximated and a trust-region mechanism is implemented.

Section 2 gives a short summary of the UHVI and the linear-quadratic rank-oriented surrogate. The implementation of the linear-quadratic surrogate of the UHVI in the Sofomore framework is presented in Section 3. Numerical experiments are described and discussed in Section 4 and Section 5 concludes the paper.

2 PREVIOUS WORK

The algorithm introduced in this paper is an instance of the Sofomore framework with lq-CMA-ES as a single objective optimizer. To understand the underlying algorithm we first describe the Sofomore framework and the lq-CMA-ES algorithm.

2.1 Sofomore: multiobjective optimization with single objective algorithms

The Sofomore framework [21] reformulates a multiobjective optimization problem as multiple single-objective problems where each problem corresponds to a *dynamically chosen* trade-off between the objective functions. The dynamic aggregation of objectives is called Uncrowded Hypervolume Improvement (UHVI) [21]. It is defined for nondominated solutions as the hypervolume improvement. For dominated solutions it is the Euclidean distance to the nondominated region in the objective space. The UHVI allows to maximize the hypervolume of p points via optimizing p single-objective dynamic problems with p instantiations of a single objective optimization algorithm. Each single objective algorithm maximizes the hypervolume given the $p - 1$ incumbent solutions of the other $p - 1$ optimizers. One instance of Sofomore is COMO-CMA-ES [21] with CMA-ES as single objective algorithm.

2.2 Surrogating CMA-ES

The lq-CMA-ES [11] is a surrogate assisted variant of CMA-ES that uses a linear or quadratic model to approximate the values of the objective function, in our case the uncrowded hypervolume

Algorithm 1 The COMO-lq-CMA-ES: an instance of the Sofomore framework with the lq-CMA-ES as single-objective optimizer

```

1: Required:
2:   objective functions  $f = (f_1, \dots, f_k)$  in dimension  $d$ 
3:   initialSampler() returns initial solutions  $\in \mathbb{R}^d$ 
4:   number of desired solutions  $p$ 
5:   global initial step-size  $\sigma_0$  for all CMA-ES
6:   fixed reference point  $r \in \mathbb{R}^k$  for the hypervolume indicator
7:   a UHVI $_r$ () function
8: Initializations:
9:  $x^{(i)} = \text{initialSampler}()$ ,  $1 \leq i \leq p$ 
10: evaluate  $f(x^{(1)}), \dots, f(x^{(p)})$ 
11:  $\text{es}(i) \leftarrow (\mu/\mu, \lambda)$ -CMA-ES( $x^{(i)}, \sigma_0$ ),  $1 \leq i \leq p$ 
12:  $\text{lq}(i) \leftarrow \text{lq.initialize}(\text{UHVI}_r, x^{(1)}, f(x^{(1)}), \dots, x^{(p)}, f(x^{(p)}))$ ,
    $1 \leq i \leq p$ 
13: while not stopping criterion do
14:   sample at random a permutation  $\pi$  of  $\{1, \dots, p\}$ 
15:   for  $i = 1$  to  $p$  do
16:      $\{s_1, \dots, s_\lambda\} \leftarrow \text{es}(\pi(i)).\text{ask}()$  # get  $\lambda$  offspring from
   #  $\pi(i)$ th CMA-ES
17:      $\text{lq}(\pi(i)).\text{update}(s_1, \dots, s_\lambda)$  # calls between 1 and  $\lambda$  vec-
   # tors of objective functions
   #  $f$  through UHVI $_r$ 
18:     compute the fitness  $\Phi(s_j) = \text{lq}(\pi(i)).\text{evaluate}(s_j)$ ,
    $1 \leq j \leq \lambda$ 
19:      $\text{es}(\pi(i)).\text{tell}(\Phi(s_1), \dots, \Phi(s_\lambda))$ 
20:      $x^{(\pi(i))} \leftarrow \text{es}(\pi(i)).\text{mean}$ 
21:     update the stored objective vector  $f(x^{(\pi(i))})$ 
22:     add  $(x^{(\pi(i))}, f(x^{(\pi(i))}))$  to the data queue  $\mathcal{M}$ 
   of  $\text{lq}(\pi(i))$ 
23: return  $x^{(1)}, \dots, x^{(p)}$ 

```

improvement (UHVI) which maps solutions from the search space either to their contributing hypervolume or to their f -distance to the nondominated domain (see above). Because CMA-ES requires only the ranking of solutions w.r.t. to the objective function, the surrogate model building and management is based on its ability to rank solutions and the lq-CMA-ES uses the *Kendall- τ* to monitor the model's quality. The lq-CMA-ES can, at least in principle, be used for the dynamic and composite UHVI without any essential algorithm changes, as described in the following section.

3 INTEGRATING LQ-CMA-ES INTO SOFOMORE

Integrating lq-CMA-ES into the Sofomore framework is straightforward, as we mainly replace each single-objective CMA-ES in COMO-CMA-ES with the lq-CMA-ES as single-objective optimization algorithm.

Using the single-objective lq-CMA-ES of [11] within the Sofomore framework [21] results in the surrogate-assisted multiobjective COMO-lq-CMA-ES algorithm. Algorithm 1 shows its pseudocode using an ask-and-tell interface. In the following, we describe its components in more detail.

Basic structure. In the Sofomore framework, p single-objective optimization algorithms optimize the multiobjective problem concurrently in the sense that their p incumbents aim at approximating the optimal p -distribution of the hypervolume indicator [2]. Henceforth, single-objective optimization algorithms will also be called kernels (they represent in our case also a unimodal probability density over the input space whose modal and mean value is the kernel's incumbent). Using a single-objective algorithm (a kernel) with an ask-and-tell interface [14], the algorithm boils down to a few basic steps: after an initial sampling (lines 9–10) and the initialization of the p kernels (line 11) and surrogates (line 12), the COMO-lq-CMA-ES advances each of the p kernels in a random order by one ask-and-tell step each. The objective function is given by the linear-quadratic surrogate to the UHVI of the entire set of incumbents without the incumbent of the currently advancing kernel—thus dynamically changing from one iteration to the next because the incumbents of the other kernels changed in the meantime. The objective function is called during the updating of the surrogate, line 17. In addition to the evaluations of the incumbents on the objective function (line 21), there are between one and λ (the population size) calls to the true f at each iteration of each CMA-ES optimizer. The exact number of calls to f depends on the *Kendall- τ* test within the `lq.update` function, which is depicted, along with other functionalities of the `lq` surrogate in Algorithm 2. At least 2% of the population is evaluated at the first *Kendall- τ* test. The incumbent data points are added to the data queue of their corresponding *lq* model (line 22 in Algorithm 1), which is not the case in the original algorithm [11]. Additional details (e.g., the rate of growth of the number of evaluated individuals) can be found in [11].

Surrogate updates account for the dynamic function. The UHVI is a dynamic function, changing with the position of the (other) incumbents (empirical Pareto front). This is readily taken care of by *i*) recalculating at each iteration the UHVI of all the data points in \mathcal{M} , and *ii*) refitting the quadratic surrogate afterwards, as done in `lq.update`, Algorithm 2. This is feasible thanks to the numerical efficiency of this surrogate whose computational cost is that of a pseudo-inverse of a matrix of size the number of parameters of the surrogate, for a total cost of $\mathcal{O}(d^6)$ ($\mathcal{O}(d^2)$ parameters in the quadratic response surface and the cost of the pseudo-inverse is cubic). Additionally, recalculating the (contributing) hypervolume of k kernels takes only $\Theta(k \log k)$ for two objectives, which is negligible compared to model building.

Modeling capacity of the quadratic surrogate. The quadratic surrogate can correctly predict any twice continuously differentiable function in a sufficiently small neighborhood around a point. This is not the case everywhere of the UHVI function (cf. Figure 1 top right in [21]) and no trust region control is included in `lq-CMA-ES` since the step size σ evolves according to the CMA-ES dynamics. However, when the quadratic surrogate does not correctly predict the ranks, all λ points of the population end up being evaluated with the true functions f and the surrogate is not used at this iteration. In other words, when the quadratic surrogate does not work, `lq-CMA-ES` degenerates into CMA-ES (see `lq.update` and `lq.evaluate` functions in Algorithm 2).

Algorithm 2 The linear-quadratic surrogate functionalities within `lq-CMA-ES`

Required:

UHVI $_r$ (), a function which uses the reference point r and a set of p vectors of objective functions, $f(x^{(1)}), \dots, f(x^{(p)})$

function `LQ.INITIALIZE`(UHVI $_r$, $x^{(1)}, f(x^{(1)}), \dots, x^{(p)}, f(x^{(p)})$)
 start a data queue \mathcal{M} made of sorted $(x^{(i)}, f(x^{(i)}))$,
 $i = 1, \dots, p$, the best last in queue
return a quadratic surrogate for `lq-CMA-ES` approximating the UHVI with r as reference point and the empirical Pareto front extracted from $f(x^{(1)}), \dots, f(x^{(p)})$

function `LQ.UPDATE`(s_1, \dots, s_λ)
 recalculate UHVI $_r$ () for all points in the queue \mathcal{M} and refit the surrogate # because the Pareto front may have # changed (without new call to f)
while *Kendall- τ* < 0.85 **and** not all population evaluated **do**
 sort s_1, \dots, s_λ according to `lq.evaluate`(s_i), $i = 1, \dots, \lambda$
 evaluate UHVI $_r$ () of a small, increasing fraction of them, starting with the surrogate bests # (UHVI $_r$ (s_i)) calls # the true f (s_i)
 calculate *Kendall- τ* between the true and surrogated UHVI of the fraction of individuals
 add the newly evaluated points to the queue \mathcal{M} , the best last in queue. Remove oldest individuals from \mathcal{M} until its size is $\max(\lambda, 2 \times \text{number of surrogate parameters})$
 refit the quadratic surrogate to \mathcal{M}
return `lq`

function `LQ.EVALUATE`(s)
if $s \in$ last population **and** all λ individuals evaluated **then**
return UHVI $_r$ () at s
else
return surrogate prediction for UHVI $_r$ () at s

4 NUMERICAL EXPERIMENTS

In this section, we show experiments on the double sphere function and on the `bbob-biobj` test suite [5] of the COCO benchmarking platform [12].

4.1 Double Sphere Function

We investigate the convergence behavior of COMO-lq-CMA-ES on the double sphere function.

4.1.1 Experimental setup. We run experiments with COMO-CMA-ES and COMO-lq-CMA-ES with $p = 2, 10, 30$ on the double sphere function

$$f_1(x) = \sum_{i=1}^d x_i^2, \quad f_2(x) = (x_1 - 3)^2 + \sum_{i=2}^d x_i^2 \quad (1)$$

with the reference point at (10, 10). We initialize each x_0 uniformly at random in $[-5, 5]^d$ and set the initial step size $\sigma_0 = 2/\sqrt{d}$. The termination conditions `tolfun`, `tolfunhist`, `tolstagnation`, and `tolx` were set to 0 to avoid early termination within the displayed

budget. We use the parameter settings for lq-CMA-ES as in [11] (in particular *Kendall- τ* threshold of 0.85 and the minimum number of function evaluations per iteration of 2%). The population size of the kernels is set to a `popsize_factor` \times $(4 + 3 \times \ln(d))$ with `popsize_factor` = 1 and 4.

4.1.2 Single runs. Figure 1 shows plots from two single runs on the 20-dimensional double sphere function with two kernels.

In addition to the plots from the `cma` Python module, we monitored also the performance of the surrogate model. The surrogate fitness values as used in the selection of lq-CMA-ES and plotted by the `cma` module are in itself uninformative. Hence we show in the upper right subfigures the UHVI of the distribution mean of the lq-CMA-ES when the second kernel is at the optimum, as the gap to its value when both kernels are at the optimum. This graph and the convergence of the variables (upper middle) and the step-size σ indicate linear convergence to the optimum. As the second kernel exhibits the same behavior for the UHVI (not shown), the optimal hypervolume of 72 is approached. Apart from a subtle dip around 100 and 50 iterations, respectively for population size 12 and 51, the dynamic nature of the fitness seems to have no visible impact on the behavior while the convergence rate *per iteration* is about 10-20% slower than for CMA-ES on the sphere function. The dip is much more pronounced in the *Kendall- τ* correlation between the surrogate and the true UHVI values for the population (green and black lines in the lower right plot) indicating a comparatively poor model fit. This could be due to a fast moving kernel or to the difficulty to model the composite UHVI on the boundary between domination and nondomination.

Peculiarly, the *Kendall- τ* 's drop to zero for the final 50-100 iterations and consequently all solutions are evaluated during these iterations. We attribute this effect to numerical precision problems in the model building process.

The principal axis of the sample ellipsoid (lower left) and the standard deviations in the coordinates (lower middle) stay within a factor of about 2-3. The standard deviation for the first variable is about two times smaller, indicating that the sensitivity along the Pareto set is slightly higher than in the perpendicular directions. Overall, this suggests that the UHVI landscape for the double sphere function is relatively benign.

4.1.3 Performance effects from the surrogate. Figure 2 shows the progression of the gap between the hypervolume achieved by all kernels to the largest observed hypervolume (in the subplot) for COMO-lq-CMA-ES and COMO-CMA-ES on the double sphere function. Both algorithms exhibit, after an initial phase, linear convergence up to apparent numerical precision limits. The convergence rate of COMO-lq-CMA-ES is about six and 20 times faster than that of COMO-CMA-ES with the default and $4 \times$ default population size, respectively. Below a gap of 10^{-10} (and 10^{-8} with 32 kernels), the convergence rate of COMO-lq-CMA-ES drops back to the rate of COMO-CMA-ES, because all solutions are evaluated (see above).

4.2 COCO

We use the biobjective test suite [5] of the COCO benchmarking framework [12] to evaluate the effect of the introduced surrogate model building on a wide variety of functions.

4.2.1 Experimental setup. We run experiments on the bbob-biobj test function suite from COCO in dimensions 2, 3, 5, 10, 20. We used the same setup as in the previous subsection for the double sphere function for both algorithms. The reference point is provided for each problem by its attribute `largest_fvalues_of_interest`. We run the experiments until the budget of $10,000 \times d$ function evaluations is exhausted without restarts. Moreover, with our current settings we only hit the maximum number of function evaluations termination condition.

The performance assessment is done in two different ways: (i) by the COCO standard assessment based on all evaluated solutions to compute the hypervolume and (ii) by using only the incumbents of the algorithms for the hypervolume computation. Specifically, we evaluate only the incumbent(s) of the advancing kernel(s) on the observed problem as often as the number of function evaluations used for the model building. We adopt this approach to reduce the effects from random sampling since we optimize for the p -optimal distribution on the Pareto front. Both results are presented in this paper. The assessment from incumbents is distinguished by adding the ker prefix to the name of the algorithm.¹

Statistical significance test results are those provided by the COCO platform (rank sum tests).

4.2.2 Results. Figure 3 shows overall empirical runtime distributions for the kernel hypervolume of ten kernels (orange and blue) and hundred kernels (red and green) as quality measure of COMO-lq-CMA-ES and COMO-CMA-ES.

For ten kernels, the COMO-lq-CMA-ES (blue) is in all dimensions overall about 25% faster than COMO-CMA-ES (orange). For 100 kernels, the COMO-lq-CMA-ES (green) is in all dimensions overall two to three times faster than COMO-CMA-ES (red).

Figures 4 and 5 show the results on the single functions in dimension 20. The COMO-lq-CMA-ES is statistically significantly better than the other three variants for at least one of the five target precisions $1, 10^{-1}, 10^{-2}, 10^{-3}$ and 10^{-4} on functions 1–6, 9, 12, 21, 28, 29, 33, 35, 36, 39, 46, 53 for ten kernels and on functions 1, 4, 5, 6, 9, 22, 29, 35, 41, 46 for 100 kernels.

We also find the COMO-lq-CMA-ES with 10 kernels to be statistically significantly better than the other three variants for at least one target on 1, 0, 3, 11 and 17 of the 55 functions in dimension 2, 3, 5, 10 and 20, respectively. With 100 kernels, COMO-lq-CMA-ES is statistically significantly better than the other three variants on 1, 3, 8, 10 and 10 of the 55 functions in dimension 2, 3, 5, 10 and 20, respectively, for at least one target. On the other hand, COMO-CMA-ES was never observed to be statistically significantly better than the other three variants.

Figure 6 shows empirical runtime distributions aggregated over all functions where, as by default, all evaluated solutions are taken into account for the performance assessment (see Section 4.2.1). Apart from our results with COMO-CMA-ES and COMO-lq-CMA-ES, we show results for MO-CMA-ES [15, 22] and COMO-CMA-ES (as COMO-100), both benchmarked in [10], and for NSGA-III [8] as benchmarked in [6] and denoted by N-III-111.² The shown algorithms have a population size of 100 except for N-III-111 with a

¹All experimental COCO data are at <https://mgharafi.github.io/COMO-lq-CMA-ES>.

²The data sets (COMO-100_dufosse, MO-CMA-ES-100, NS-III-111-platypus) are taken from <https://numbbo.github.io/data-archive/data-archive/bbob-biobj/2019>.

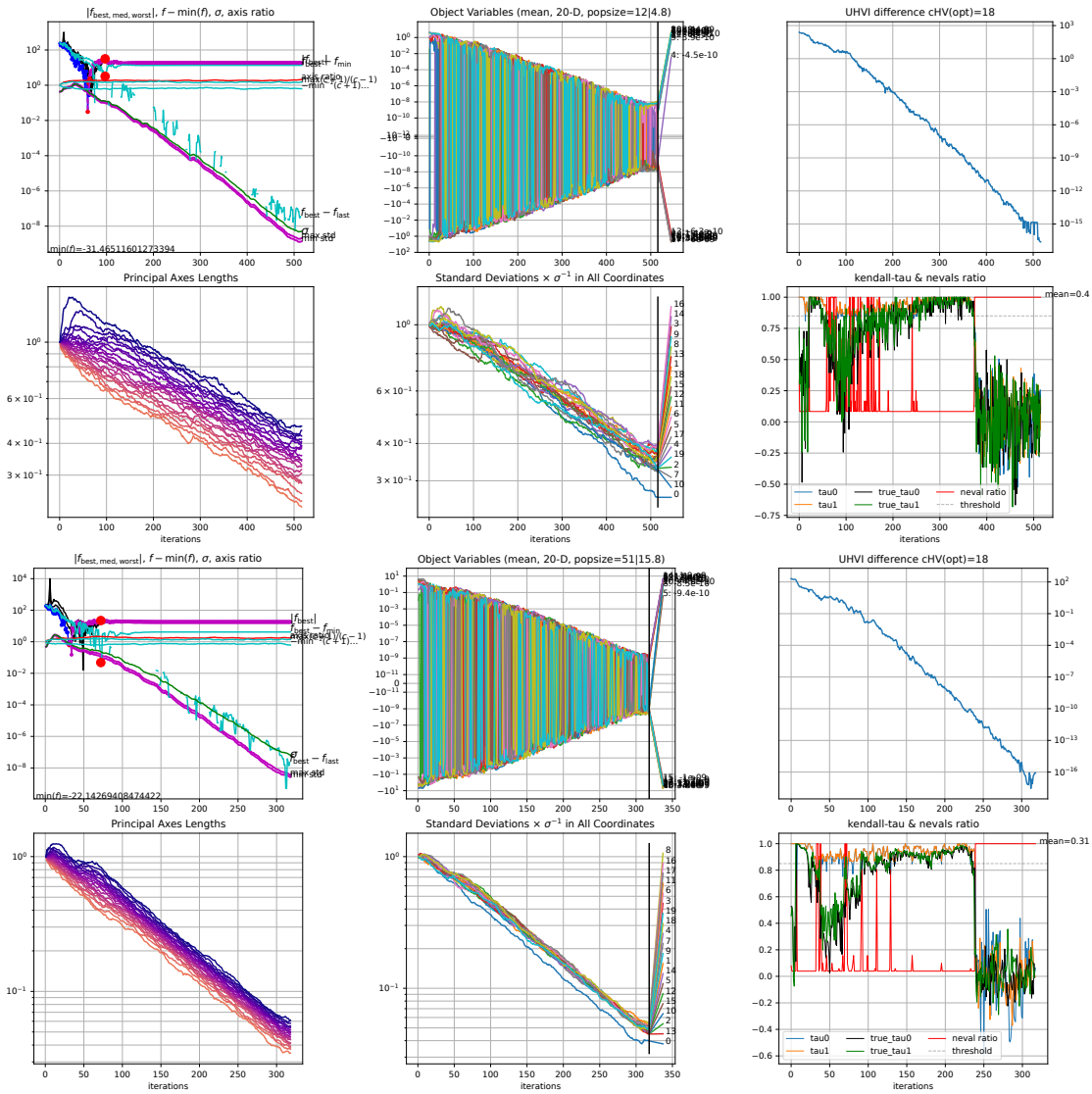


Figure 1: Two single runs of COMO-lq-CMA-ES on the double sphere function in dimension 20 with $p = 2$ and population size $\lambda = 12$ (above) and 51 (below). Shown are quantities related to one of the two kernels. The upper left figure shows, per iteration, the absolute value of the population-best surrogate value (blue, negative values in magenta), its difference to the overall minimum value (cyan), its difference between two iterations (cyan with gaps), the step-size σ (green), and the largest and smallest sample standard deviation per coordinate (magenta). The lower left figure shows the square root of the eigenvalues of the covariance matrix (principal axes lengths). The upper middle figure shows for each variable the difference to the optimal value in symlog scale. The lower middle figure shows the sample standard deviations in each coordinate divided by the step-size σ . The upper right figures display the known UHVI at the optimum minus the UHVI of the kernel given the other kernel is at the optimum. The lower right figure displays the ratio of evaluated solutions (red) and various *Kendall*- τ 's. The measurements τ_{00} and τ_{11} are those used to decide on making further f -evaluations in lq-CMA-ES after, respectively, the first and the last evaluation of the iteration [11]. The measurements τ_{01} and τ_{10} give the ranking mismatch between surrogate values and true UHVI values of the population, respectively before the first and after the last evaluation of the iteration.

population size of 111. The notable difference between our COMO-CMA-ES experiment and the data of COMO-100_dufosse are due to the $d/2$ times larger initial step-size in the latter data (as we have confirmed in additional experiments). Overall, similar to the

previous observations, COMO-lq-CMA-ES is about 25% faster than COMO-CMA-ES in dimension 3 (above 500 d evaluations), the performance gap increases with increasing dimension, and it becomes

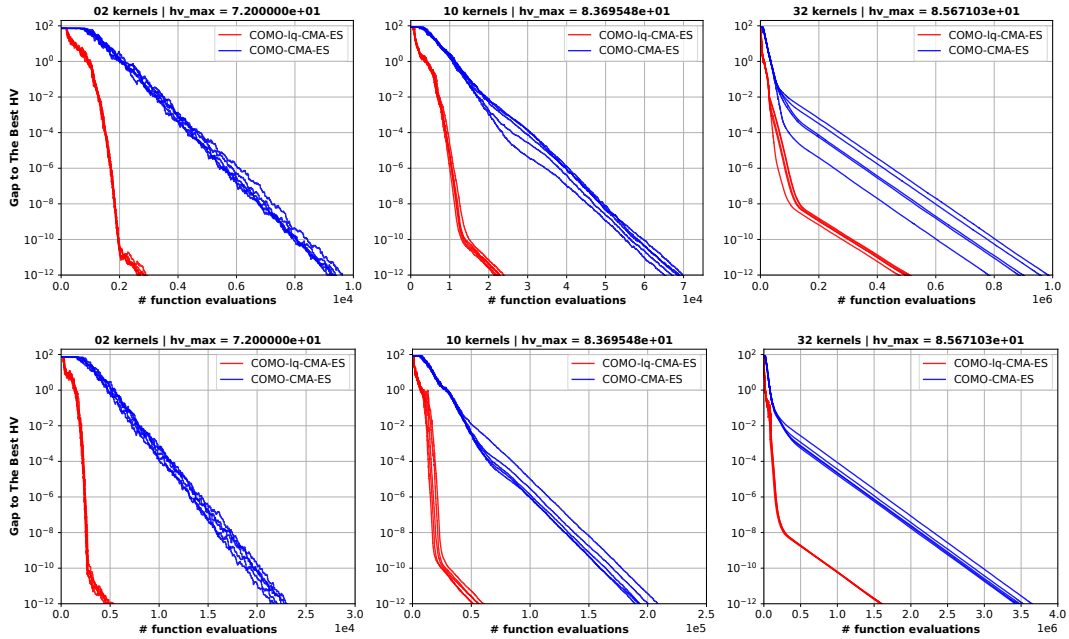


Figure 2: Hypervolume gap on the double sphere function of COMO-lq-CMA-ES (red) and COMO-CMA-ES (blue) with `popsize_factor=1` (top, 5 runs) and `popsize_factor=4` (bottom, 5 runs) and, from left to right, 2, 10 and 32 kernels. The hypervolume gap is computed as the difference to the largest hypervolume observed within each subplot.

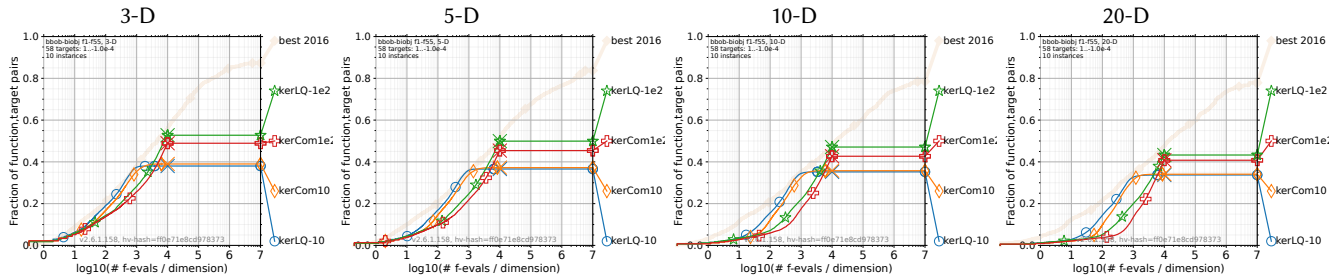


Figure 3: Empirical cumulative distribution of the number of f -evaluations divided by dimension (FEvals/DIM) for 58 targets with target precision in $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \dots, 10^{-0.1}, 10^0\}$ for all functions in dimensions 3, 5, 10, and 20-D. As reference algorithm, the artificial best algorithm from BBOB 2016 is shown as light thick line with diamond markers.

between a factor of about 1.5 and four in dimension 20 where the gap diminishes with increasing runtimes.

Up to a budget of somewhere above $1000 \times d$ evaluations, NS-III is the best algorithm in all dimensions, then overtaken by MO-CMA-ES, and above around $5 \times 10^4 d$ evaluations COMO-lq-CMA-ES becomes the best algorithm except in dimension 3.

5 SUMMARY AND CONCLUDING REMARKS

In this paper, we have added the linear-quadratic surrogate model building procedure as it was introduced in the lq-CMA-ES [11] into the COMO-CMA-ES [21]. Even though the CMA-ES must optimize a dynamically changing fitness within COMO, the lq-CMA-ES does

not require any *additional* model building effort when integrated in COMO.

Our experiments do not reveal any particular flaws or failures from integrating the linear-quadratic surrogate model into COMO-CMA-ES. On the double sphere function, the COMO-lq-CMA-ES achieves a six times faster convergence rate than COMO-CMA-ES with default population size, and 20 times faster with a four times larger population size. On the bbo-biobj test suite from COCO, a saving of 20-60% of the function evaluations is achieved on average. These improvements are however smaller than those observed in the single-objective case [11], in particular in smaller dimensions or with a small number of kernels or with a larger

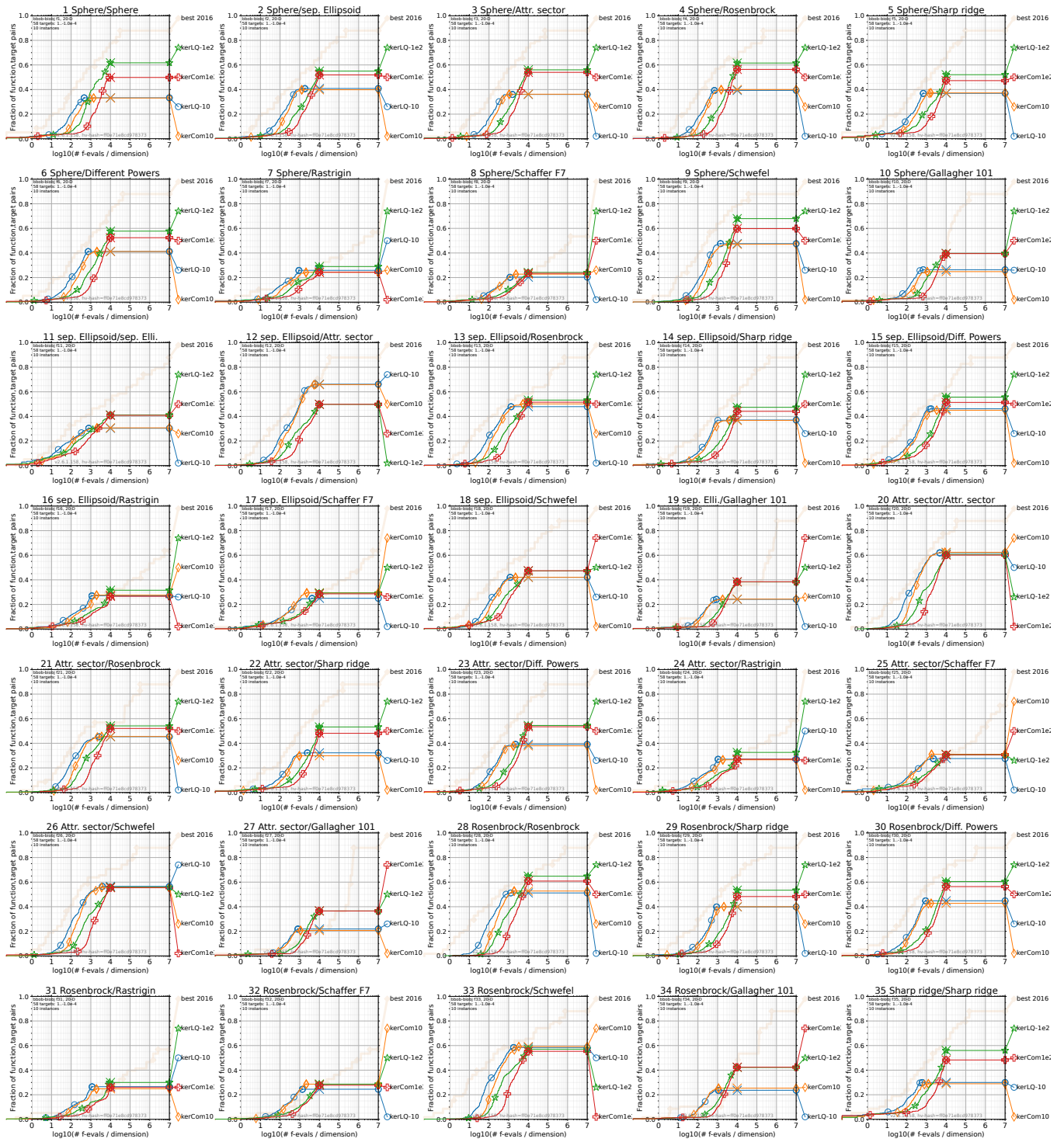


Figure 4: Empirical cumulative distribution of runtimes, measured in number of f -evaluations, divided by dimension (FEvals/DIM) for the 58 targets $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \dots, 10^{-0.1}, 10^0\}$ in dimension 20.

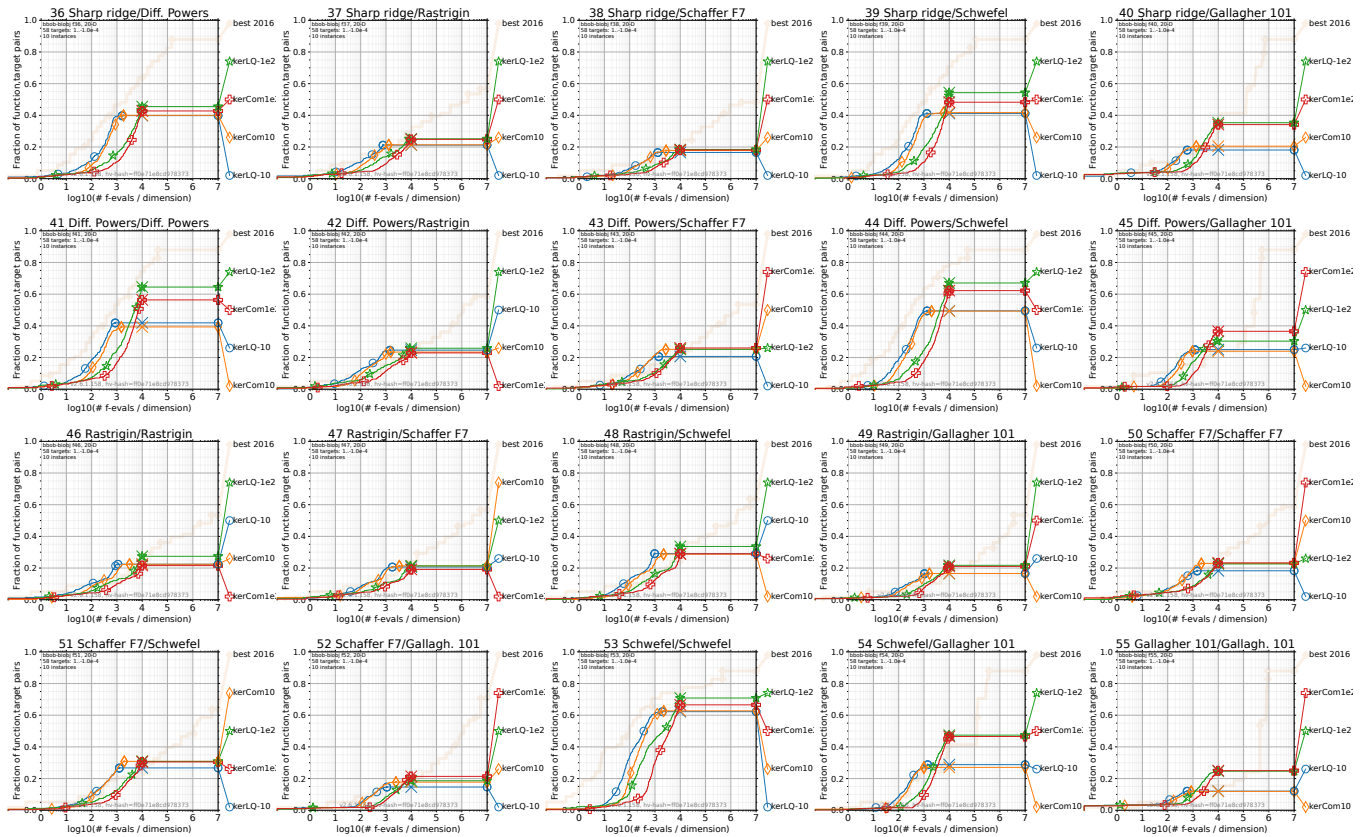


Figure 5: Empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) as in Fig. 4 but for functions f_{36} to f_{55} in 20-D.

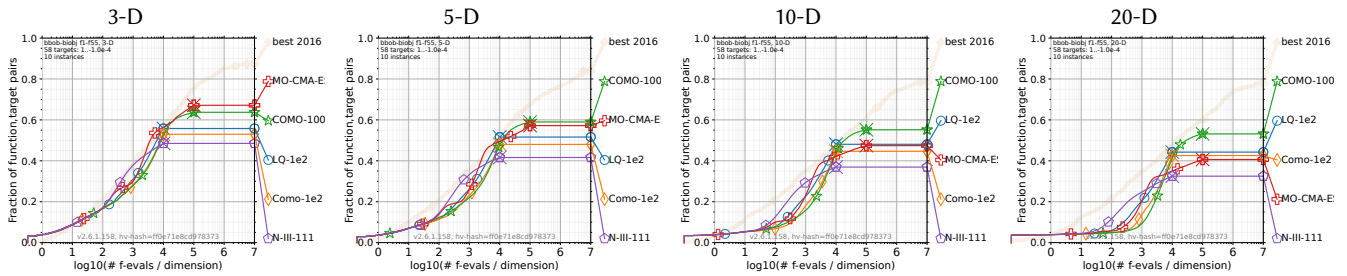


Figure 6: Empirical cumulative distribution of the number of f -evaluations divided by dimension (FEvals/DIM) for 58 targets with target precision in $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \dots, 10^{-0.1}, 10^0\}$ for all functions in 3, 5, 10 and 20-D. As reference algorithm, the artificial best algorithm from BBOB 2016 is shown as light thick line with diamond markers.

budget. While the advantage of COMO-lq-CMA-ES is on many functions statistically significant, we observe no case where COMO-CMA-ES is statistically significantly better than COMO-lq-CMA-ES. All our experiments were conducted with the original default parameter settings *without any additional parameter tuning*.

ACKNOWLEDGEMENTS

This research was conducted with the support of the consortium in Applied Mathematics CIROQUO, gathering partners in technological and academia for the development of advanced methods for Computer Experiments.

REFERENCES

- [1] Richard Allmendinger, Michael TM Emmerich, Jussi Hakanen, Yaochu Jin, and Enrico Rigoni. 2017. Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case. *Journal of Multi-Criteria Decision Analysis* 24, 1-2 (2017), 5–24.
- [2] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. 2009. Theory of the hypervolume indicator: optimal μ -distributions and the choice of the reference point. In *Foundations of Genetic Algorithms (FOGA 2009)*. ACM, New York, NY, USA, 87–102.
- [3] Mickaël Binois and Victor Picheny. 2019. GPareto: An R package for gaussian-process-based multi-objective optimization and analysis. *Journal of Statistical Software* 89 (2019), 1–30.
- [4] G. E. P. Box and K. B. Wilson. 1951. On the Experimental Attainment of Optimum Conditions. *Journal of the Royal Statistical Society: Series B (Methodological)* 13, 1 (1951), 1–38. <https://doi.org/10.1111/j.2517-6161.1951.tb00067.x> arXiv:<https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1951.tb00067.x>
- [5] Dimo Brockhoff, Anne Auger, Nikolaus Hansen, and Tea Tušar. 2022. Using well-understood single-objective functions in multiobjective black-box optimization test suites. *Evolutionary Computation* 30, 2 (2022), 165–193.
- [6] Dimo Brockhoff and Tea Tušar. 2019. Benchmarking algorithms from the platypus framework on the biobjective bbob-biobj testbed. In *Genetic and Evolutionary Computation Conference Companion (GECCO 2019)*. ACM, New York, NY, USA, 1905–1911.
- [7] Tinkle Chugh, Karthik Sindhya, Jussi Hakanen, and Kaisa Miettinen. 2019. A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Computing* 23 (2019), 3137–3166.
- [8] Kalyanmoy Deb and Himanshu Jain. 2013. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (2013), 577–601.
- [9] Kalyanmoy Deb, Proteek Chandan Roy, and Rayan Hussein. 2020. Surrogate modeling approaches for multiobjective optimization: methods, taxonomy, and results. *Mathematical and Computational Applications* 26, 1 (2020), 5.
- [10] Paul Dufossé and Cheikh Touré. 2019. Benchmarking MO-CMA-ES and COMO-CMA-ES on the Bi-objective bbob-biobj Testbed. In *Genetic and Evolutionary Computation Conference Companion (GECCO 2019)*. ACM, New York, NY, USA, 1920–1927.
- [11] Nikolaus Hansen. 2019. A global surrogate assisted CMA-ES. In *Genetic and Evolutionary Computation Conference (GECCO 2019)*. ACM, New York, NY, USA, 664–672.
- [12] 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.
- [13] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* 9, 2 (2001), 159–195.
- [14] Nikolaus Hansen, Gilles Pujol, Daniel Salazar Aponte, and Rodolphe Le Riche. 2010. On Object-Oriented Programming of Optimizers-Examples in Scilab. In *Multidisciplinary Design Optimization in Computational Mechanics*, Piotr Breitung and Rajan Filomeno Coehlo (Eds.). Wiley, Hoboken, NJ, USA, 499–538.
- [15] Christian Igel, Nikolaus Hansen, and Stefan Roth. 2007. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation* 15, 1 (2007), 1–28.
- [16] Marios K. Karakasis and Kyriakos Giannakoglou. 2005. Metamodel-assisted multi-objective evolutionary optimization. In *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN 2005)*, R. R. Schilling, W. Haase, J. Periaux, H. Baier, and G. Bugeda (Eds.). Springer, Berlin/Heidelberg, Germany.
- [17] GP Liu, Xu Han, and Chao Jiang. 2008. A novel multi-objective optimization method based on an approximation model management technique. *Computer Methods in Applied Mechanics and Engineering* 197, 33-40 (2008), 2719–2731.
- [18] Ilya Loshchilov, Marc Schoenauer, and Michèle Sebag. 2010. A mono surrogate for multiobjective optimization. In *Genetic and Evolutionary Computation Conference (GECCO 2010)*. ACM, New York, NY, USA, 471–478.
- [19] Martin Pilát and Roman Neruda. 2015. Hypervolume-based surrogate model for MO-CMA-ES. In *International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE Computer Society, Washington, D.C., USA, 604–611.
- [20] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. 2015. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* 104, 1 (2015), 148–175.
- [21] Cheikh Touré, Nikolaus Hansen, Anne Auger, and Dimo Brockhoff. 2019. Uncrowded hypervolume improvement: COMO-CMA-ES and the Sofomore framework. In *Genetic and Evolutionary Computation Conference (GECCO 2019)*. ACM, New York, NY, USA, 638–646.
- [22] Thomas Voß, Nikolaus Hansen, and Christian Igel. 2010. Improved step size adaptation for the MO-CMA-ES. In *Conference on Genetic and Evolutionary Computation (GECCO 2010)*. ACM, New York, NY, USA, 487–494.