

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/332511718>

Uncrowded hypervolume improvement: COMO-CMA-ES and the sofomore framework

Conference Paper · April 2019

DOI: 10.1145/3321707.3321852

CITATIONS
23

READS
196

4 authors, including:



Cheikh Toure
École Polytechnique
14 PUBLICATIONS 51 CITATIONS

[SEE PROFILE](#)



Nikolaus Hansen
Inria (National Institute for Research in Computer Science and Control)
220 PUBLICATIONS 29,204 CITATIONS

[SEE PROFILE](#)



Anne Auger
University of Paris-Sud
185 PUBLICATIONS 11,227 CITATIONS

[SEE PROFILE](#)

Uncrowded Hypervolume Improvement: COMO-CMA-ES and the Sofomore framework

Cheikh Touré

Inria, CMAP, Ecole Polytechnique, IP Paris
cheikh.toure@polytechnique.edu

Anne Auger

Inria, CMAP, Ecole Polytechnique, IP Paris
firstname.lastname@inria.fr

ABSTRACT

We present a framework to build a multiobjective algorithm from single-objective ones. This framework addresses the $p \times n$ -dimensional problem of finding p solutions in an n -dimensional search space, maximizing an indicator by dynamic subspace optimization. Each single-objective algorithm optimizes the indicator function given $p - 1$ fixed solutions. Crucially, dominated solutions minimize their distance to the empirical Pareto front defined by these $p - 1$ solutions. We instantiate the framework with CMA-ES as single-objective optimizer. The new algorithm, COMO-CMA-ES, is empirically shown to converge linearly on bi-objective convex-quadratic problems and is compared to MO-CMA-ES, NSGA-II and SMS-EMOA.

CCS CONCEPTS

- Mathematics of computing → Nonconvex optimization;

KEYWORDS

Multiobjective optimization, single-objective optimization, hypervolume, quality indicator, hypervolume contribution, hypervolume improvement

ACM Reference Format:

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 '19), July 13–17, 2019, Prague, Czech Republic*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3321707.3321852>

1 INTRODUCTION

Multiobjective optimization problems must be solved frequently in practice. In contrast to the optimization of a single objective, solving a multiobjective problem involves to handle trade-offs or incomparabilities between the objective functions such that the aim is to approximate the Pareto set—the set of all Pareto-optimal or non-dominated solutions. One might be interested to obtain an approximation of unbounded size (the more points the better)

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

© 2019 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Genetic and Evolutionary Computation Conference (GECCO '19), July 13–17, 2019, Prague, Czech Republic*, <https://doi.org/10.1145/3321707.3321852>.

Nikolaus Hansen

Inria, CMAP, Ecole Polytechnique, IP Paris
firstname.lastname@inria.fr

Dimo Brockhoff

Inria, CMAP, Ecole Polytechnique, IP Paris
firstname.lastname@inria.fr

or just to have p points approximating the Pareto set. Evolutionary Multiobjective Optimization (EMO) algorithms aim at such an approximation in a single algorithm run whereas more classical approaches, e.g. optimizing a weighted sum of the objectives with changing weights, operate in multiple runs.

The first introduced EMO algorithms simply changed the selection of an existing single-objective evolutionary algorithm keeping the exact same search operators. The population at a given iteration was then providing an approximation of the Pareto set. This idea led to the practically highly successful NSGA-II algorithm [8] that employs a two-step fitness assignment: after a first non-dominated ranking [11], solutions with equal non-domination rank are further distinguished by their crowding distance—based on the distance of each solution to its neighbors in objective space. However, it has been pointed out that NSGA-II does not converge to the Pareto set in a mathematical sense due to so-called deteriorative cycles: if all population members of the algorithm are non-dominated at some point in time, it is only the crowding distance that is optimized, without indicating any search direction towards the Pareto set to the algorithm. As a result, solutions which had been non-dominated solutions at some point in time can be replaced by previously dominated ones during the optimization, ending up in a cyclic but not in convergent behavior [3].

To improve the convergence properties of EMO algorithms, different approaches have been introduced later, most notably the indicator-based algorithms and especially algorithms based on the hypervolume indicator. They replace the crowding distance of NSGA-II with the (hypervolume) indicator contribution, see e.g. [4, 17]. Using the hypervolume indicator has the advantage that it is the only known strictly monotone quality indicator [19] (see also next section) and thus, its optimization will result in solution sets that are subsets of the Pareto set.

The optimization goal of indicator-based algorithms such as SMS-EMOA [4] or MO-CMA-ES [17] is to find the best set of p solutions with respect to a given quality indicator (the set with the largest quality indicator value among all sets of size p). This optimal set of p solutions is known as the optimal p -distribution [1]. In principle, the search for the optimal p -distribution can be formalized as a $p \cdot n$ -dimensional optimization problem where p is the number of solutions and n is the dimension of the search space.

As we will discuss later, it turns out that this optimization problem is not only of too high dimension in practice but also flat in large regions of the search space if the hypervolume indicator is the underlying quality indicator. The combination of non-dominated

ranking and hypervolume contribution as in SMS-EMOA or MO-CMA-ES corrects for this flatness, but also introduces search directions that are pointing towards already existing non-dominated solutions and not towards not-yet-covered regions of the Pareto set. In this paper, we show that we can correct the flat region of the hypervolume indicator by introducing a search bias towards yet-uncovered regions of the Pareto set by adding the distance to the empirical non-domination front, which leads to the new notion of Uncrowded Hypervolume Improvement. Then, we define a (dynamic) fitness function that can be optimized by single-objective algorithms. From there, going back to this original idea of EMO algorithms to use single-objective optimizers to build an EMO, we define the *Single-objective Optimization FOr Optimizing Multiobjective Optimization pRoblEms* framework (Sofomore) to build in an elegant manner, a multiobjective algorithm from a set of p single-objective optimizers. Each single-objective algorithm optimizes (iteratively or in parallel) a dynamic fitness that depends on the output of the other $p - 1$ optimizers.

We instantiate the Sofomore framework with the state-of-the-art single-objective algorithm CMA-ES. We show experimentally that the ensuing COMO-CMA-ES (Comma-Selection Multiobjective CMA-ES) exhibits linear convergence towards the optimal p -distribution on a wide variety of bi-objective convex quadratic functions. In contrast, default implementations of the SMS-EMOA where the reference point is fixed and NSGA-II do not exhibit this linear convergence. The comparison between COMO-CMA-ES and a previous MATLAB implementation of the elitist MO-CMA-ES also shows the same or an improved convergence speed in COMO-CMA-ES except for the double sphere function.

The paper is structured as follows. In the next section, we start with preliminaries related to multiobjective optimization and quality indicators. Section 3 discusses the fitness landscape of indicator- and especially hypervolume-based quality measures and eventually introduces our Sofomore framework. Section 4 gives details about the new COMO-CMA-ES algorithm as an instantiation of Sofomore with CMA-ES. Section 5 experimentally validates the new algorithm and compares it with three existing algorithms from the literature and Section 6 discusses the results and concludes the paper.

2 PRELIMINARIES

In the following, we assume without loss of generality the minimization of a vector-valued function $\mathbf{f} : \mathbf{x} \in \mathbb{R}^n \mapsto \mathbf{f}(\mathbf{x}) = (\mathbf{f}_1(\mathbf{x}), \dots, \mathbf{f}_k(\mathbf{x})) \in \mathbb{R}^k$ that maps a search point from the search space \mathbb{R}^n of dimension n to the objective space $\mathbf{f}(\mathbb{R}^n) \subseteq \mathbb{R}^k$. This minimization of \mathbf{f} is generally formalized in terms of the weak Pareto dominance relation for which we write that a search point $\mathbf{x} \in \mathbb{R}^n$ weakly Pareto-dominates another search point $\mathbf{y} \in \mathbb{R}^n$ (written in short as $\mathbf{x} \leq \mathbf{y}$ or with an abuse of notation as $\mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{y})$) if and only if $\mathbf{f}_i(\mathbf{x}) \leq \mathbf{f}_i(\mathbf{y})$ for all $i \in \{1, \dots, k\}$. Note also that we can naturally extend the (weak) Pareto dominance relation to subsets $A, B \subset \mathbb{R}^n$ as $A \leq B$ if and only if for all $b \in B$, there exists $a \in A$ such that $a \leq b$. If the relation \leq is strict for at least one objective function, we say that \mathbf{x} Pareto-dominates \mathbf{y} (and write $\mathbf{x} < \mathbf{y}$). The set of non-dominated search points constitutes the so-called Pareto set, its image under \mathbf{f} is called the Pareto front. In the remainder, we will also use the term *empirical non-dominated front* or *empirical*

Pareto front ($\text{EPF}_{S, \mathbf{r}}$) for objective vectors that are on the boundary of the (objective space) region dominating a reference point $\mathbf{r} \in \mathbb{R}^k$, and not dominated by any element of $\mathbf{f}(S)$ with $S \subset \mathbb{R}^n$:

$$\text{EPF}_{S, \mathbf{r}} = \partial U_{S, \mathbf{r}}, \text{ with } U_{S, \mathbf{r}} = \{\mathbf{z} < \mathbf{r}; \forall \mathbf{s} \in S, \mathbf{f}(\mathbf{s}) \not\leq \mathbf{z}\} \quad (1)$$

where $\partial U_{S, \mathbf{r}}$ is the boundary of the non-dominated region $U_{S, \mathbf{r}}$. Note that $\text{EPF}_{S, \mathbf{r}} \cap \mathbf{f}(\mathbb{R}^n)$ is the Pareto front when S contains the Pareto set.

Indicator-Based Set Optimization Problems. Pareto sets and Pareto fronts are, under mild assumptions, $k - 1$ dimensional manifolds. In practice, we are often interested in a finite size approximation of these sets with, let us say, p (≥ 1) many search points. To assess the quality of a Pareto set approximation $S \subseteq \mathbb{R}^n$, a quality indicator $\mathcal{I} : \mathbb{R}^n \rightarrow \mathbb{R}$ assigns a real valued quality $\mathcal{I}(S)$ to S . Formally speaking, this transforms the original multiobjective optimization of $\mathbf{f}(\mathbf{x})$ into the single-objective set problem of finding the so-called optimal p -distribution [2]

$$X_p^* = \arg \max_{X \subseteq \mathbb{R}^n, |X| \leq p} \mathcal{I}(X) \quad (2)$$

as the set of search points of cardinality p (or lower) with the highest indicator value among all sets of this size [1].

Natural candidates for practically relevant quality indicators are monotone or even strictly monotone indicators such as the epsilon-indicator [30], the R2 indicator [12], or the hypervolume indicator ([1, 31], still the only known strictly monotone indicator family to date). We remind that an indicator is called monotone if $A \leq B \implies \mathcal{I}(A) \geq \mathcal{I}(B)$ —or in other words, if it does not contradict the weak Pareto dominance relation. If $A < B \implies \mathcal{I}(A) > \mathcal{I}(B)$, we say that \mathcal{I} is strictly monotone.

Hypervolume, Hypervolume Contribution, and Hypervolume Improvement. Because the hypervolume indicator [1, 31] and its weighted variant is the only known strictly monotone indicator, we will later on use it as well in our framework. The hypervolume $HV_{\mathbf{r}}$ [32] of a finite set of solutions $S \subset \mathbb{R}^n$ with respect to the reference point $\mathbf{r} \in \mathbb{R}^k$ is defined as $HV_{\mathbf{r}}(S) = \lambda_k \left(\left\{ \mathbf{z} \in \mathbb{R}^k; \exists \mathbf{y} \in S, \mathbf{f}(\mathbf{y}) < \mathbf{z} < \mathbf{r} \right\} \right)$, where λ_k is the Lebesgue measure on the objective space \mathbb{R}^k and \mathbf{f} is the objective function. In the case of two objective functions, the hypervolume indicator value of p non-dominated solutions $S = \{\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(p)}\}$ with $\mathbf{f}_1(\mathbf{s}^{(1)}) \leq \mathbf{f}_1(\mathbf{s}^{(2)}) \leq \dots \leq \mathbf{f}_1(\mathbf{s}^{(p)})$ can also be written as the sum of the area of p axis parallel rectangles: $HV_{\mathbf{r}}(S) = \sum_{i=1}^p (\mathbf{f}_1(\mathbf{s}^{(i+1)}) - \mathbf{f}_1(\mathbf{s}^{(i)})) \cdot (r_2 - f_2(s^{(i)}))$; $f_1(s^{(p+1)}) \stackrel{\text{def}}{=} r_1$.

Furthermore, the hypervolume contribution $HVC_{\mathbf{r}}(\mathbf{s}, S)$ of a search point $\mathbf{s} \in \mathbb{R}^n$ to a solution set $S \subseteq \mathbb{R}^n$ with respect to the reference point $\mathbf{r} \in \mathbb{R}^k$ is the hypervolume indicator value that we lose when we remove \mathbf{s} from the set [5]: $HVC_{\mathbf{r}}(\mathbf{s}, S) = HV_{\mathbf{r}}(S) - HV_{\mathbf{r}}(S \setminus \{\mathbf{s}\})$.

Also, the Hypervolume Improvement $HVI_{\mathbf{r}}(\mathbf{s}, S)$ of a search point $\mathbf{s} \in \mathbb{R}^n$ to a finite set $S \subset \mathbb{R}^n$ with respect to the reference point $\mathbf{r} \in \mathbb{R}^k$ is defined as [9, 28]: $HVI_{\mathbf{r}}(\mathbf{s}, S) = HV_{\mathbf{r}}(S \cup \{\mathbf{s}\}) - HV_{\mathbf{r}}(S)$. Or in other words, $HVI_{\mathbf{r}}(\mathbf{s}, S)$ equals the increase in hypervolume when \mathbf{s} is added to the set S . Up to a null set $HVI_{\mathbf{r}}(\mathbf{s}, S) = HVC_{\mathbf{r}}(\mathbf{s}, S \cup \{\mathbf{s}\})$.

3 SOFOMORE: BUILDING MULTIOBJECTIVE FROM SINGLE-OBJECTIVE ALGORITHMS

Quality indicators have been introduced as a way to measure the quality of a set of objective vectors but also to define a multiobjective optimization problem as a single-objective set problem of maximizing the quality indicator as in (2). This naturally defines a single-objective $p \times n$ dimensional problem to be maximized

$$\mathcal{F} : (x_1, \dots, x_p) \in (\mathbb{R}^n)^p \mapsto \mathcal{I}(\{x_1, \dots, x_p\}) . \quad (3)$$

Because n and in particular p are typically large in practice, we usually do not attempt to solve a multiobjective optimization problem by directly optimizing (3). Nevertheless, when \mathcal{I} is the hypervolume indicator, Hernández et al. suggest to use a Newton method to directly solve (3). It assumes that \mathbf{f} is twice continuously differentiable, in which case the gradient and Hessian of \mathcal{F} can be computed analytically [16]. Yet, directly attacking (3) is not possible because dominated points have a zero sub-gradient and the Newton direction is therefore zero. Thus, Hernández et al. need to start from a set of non-dominated points, close enough to the Pareto set, which requires in practice to couple the approach with another algorithm [16].

Instead of directly optimizing (3), our proposed Sofomore framework performs iterative subspace optimization of the function \mathcal{F} and penalizes the flat landscape of \mathcal{F} in dominated regions. More precisely, the basic idea behind Sofomore is to optimize \mathcal{F} subspace- or component-wise, by iteratively fixing all but one search point $x^{(i)}$ and only optimizing the indicator with respect to $x^{(i)}$ while the other search points $X^{-i} := \{x^{(1)}, \dots, x^{(i-1)}, x^{(i+1)}, \dots, x^{(p)}\}$ are temporarily fixed. Hence we maximize the functions

$$\Phi_{\mathcal{I}, X^{-i}} : x \in \mathbb{R}^n \mapsto \mathcal{I}(\{x\} \cup X^{-i}) . \quad (4)$$

If the placement of each of the p search points $x^{(i)}$ ($1 \leq i \leq p$) is optimized iteratively by fixing a different point set each time, as we suggest in our Sofomore framework below, we are in the setup of optimizing a dynamic fitness. More details on this aspect of our Sofomore framework will be given below in Section 3.2.

3.1 A Fitness Function for Subspace Optimization

If we use as quality indicator \mathcal{I} in (4) a (strictly) monotone indicator like the hypervolume indicator, the overall fitness Φ is flat in the interior domain of regions where points are dominated. Hence, we suggest to not optimize (4) directly but to unflatten it in dominated areas of the search space without changing the optimization goal.

Any solution x that is dominated by the other points in X^{-i} will receive zero fitness Φ when we use as indicator in (4) the hypervolume indicator of the entire set $\{x\} \cup X^{-i}$ with respect to the reference point \mathbf{r} or replace it with the hypervolume improvement $HVI_{\mathbf{r}}(x, X^{-i})$ of the solution x to X^{-i} . This situation is depicted in the first column of Figure 1 where for a fixed set of six arbitrarily chosen search points, the hypervolume improvement's level sets (of equal fitness) in both search and objective space are shown. This flat fitness with zero gradient will not allow to steer the search towards better search points which has also been highlighted by Hernández et al. [16].

A common approach to guide an optimization algorithm in the dominated space is to use the hypervolume (contribution) as secondary fitness after non-dominated sorting [11], as it is done for example in the SMS-EMOA [4] or the MO-CMA-ES [17]. The idea is that all search points with a worse non-domination rank get assigned a fitness that is worse than for search points with a better non-domination rank. Within a set of the same rank, the hypervolume contribution with respect to all points with the same rank is used to refine the fitness. The middle column of Figure 1 shows the resulting level sets of equal fitness. As we can see, this fitness assignment distinguishes between dominated solutions, i.e. the fitness is not flat anymore. Yet it still has another major disadvantage: the search direction in the dominated area (perpendicular to its level sets) points towards already existing non-dominated solutions. Attracting dominated solutions towards non-dominated solutions seems however undesirable, as they will compete for the same hypervolume area. Instead, we want dominated points to enter the uncrowded space *between* non-dominated points thereby complementing their hypervolume contribution (improvement).

Uncrowded Hypervolume Improvement. For this purpose, we define the Uncrowded Hypervolume Improvement UHVI based on the Hypervolume Improvement for non-dominated search points and on the Euclidean distance to the non-dominated region for dominated search points. More concretely, $UHVI_{\mathbf{r}}(\mathbf{s}, S)$ of a search point $\mathbf{s} \in \mathbb{R}^n$ with respect to a finite set $S \subset \mathbb{R}^n$ and the reference point $\mathbf{r} \in \mathbb{R}^k$ is defined as

$$UHVI_{\mathbf{r}}(\mathbf{s}, S) = \begin{cases} HVI_{\mathbf{r}}(\mathbf{s}, S) & \text{if } EPF_{S, \mathbf{r}} \not\prec \mathbf{f}(\mathbf{s}) \\ -d_{\mathbf{r}}(\mathbf{s}, S) & \text{if } EPF_{S, \mathbf{r}} \prec \mathbf{f}(\mathbf{s}) \end{cases} , \quad (5)$$

where $d_{\mathbf{r}}(\mathbf{s}, S) = \inf_{y \in EPF_{S, \mathbf{r}}} d(\mathbf{f}(\mathbf{s}), y)$ is the distance between an objective vector $\mathbf{f}(\mathbf{s}) \in \mathbb{R}^k$ and the empirical non-domination front of the set S defined as in (1).

We define the fitness $\Phi_{UHVI, X^{-i}}(x)$ for a search point $x \in \mathbb{R}^n$ with respect to other solutions in X^{-i} as

$$\Phi_{UHVI, X^{-i}}(x) = UHVI_{\mathbf{r}}(x, X^{-i}) . \quad (6)$$

Note that $\Phi_{UHVI, X^{-i}}$ is continuous on the empirical non-domination front where both the hypervolume improvement and the considered distance are zero.

Figure 2 illustrates this fitness for one non-dominated and two dominated search points (blue plusses) with respect to a set of six other search points (black crosses). The right-hand column of Figure 1 shows the level sets of this fitness. The newly introduced hypervolume improvement and distance based fitness Φ_{UHVI} shows smooth level sets, both in search and in objective space. Maybe most importantly, in the dominated area, the fitness function's descent direction (perpendicular to its level sets) now points towards the gaps in the current Pareto front approximation.

3.2 Iteratively Optimizing the Φ_{UHVI} Fitness: The Sofomore Framework

After we have discussed a fitness assignment that looks worth to optimize, we come back to our initial idea of subspace optimization and define the underlying algorithmic framework behind Sofomore.

At first, we consider a *single-objective* optimizer in an abstract manner as an iterative algorithm with state $\theta \in \Theta_n$ updated as

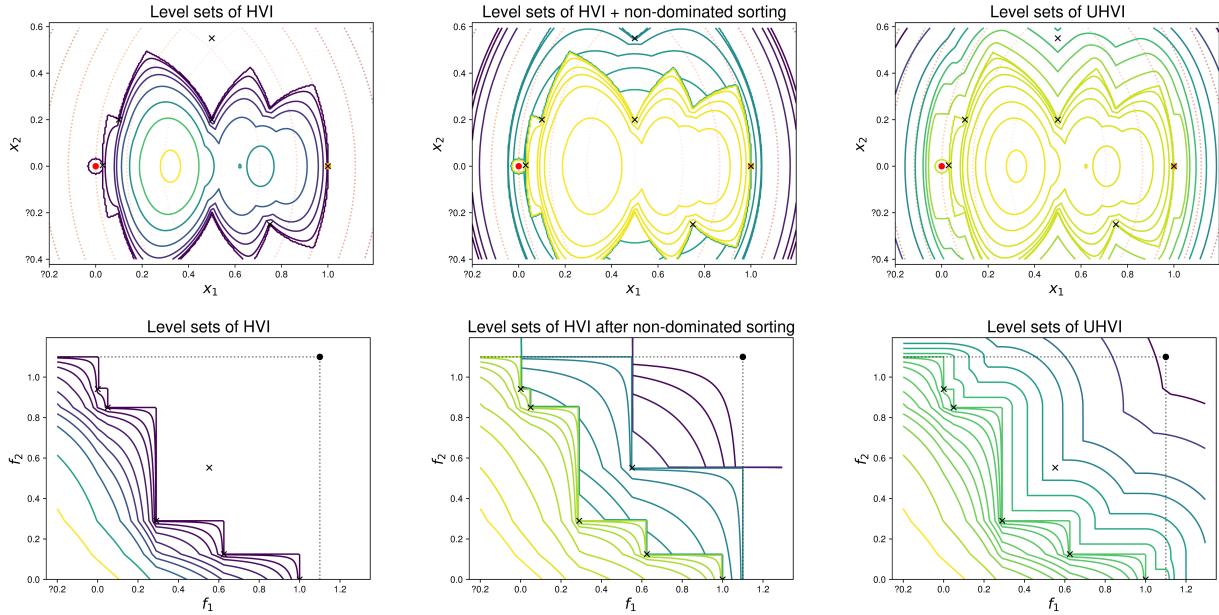


Figure 1: Comparison of block-coordinate-wise fitness functions on the double sphere problem. Given a fixed set of six solutions, $\{[0.5, 0.2], [0.75, -0.25], [0.1, 0.2], [1, 0], [0.03, 0.004], [0.5, 0.55]\}$, the above three plots show level sets of equal fitness for a new search point in the search space, the second row shows the same level sets in objective space. Left: standard hypervolume improvement HVI. Middle: HVI within the local non-dominated fronts. Right: newly proposed hypervolume improvement (if non-dominated) together with distance to the non-dominated front (if dominated), denoted as uncrowded hypervolume improvement UHVI. Note that the colors for the fitness levels are not comparable over indicators, but are the same for a given indicator in both search and objective space. The search space plots further show the single-objective's level sets as dotted lines. The black dot indicates the reference point of $[1.1, 1.1]$.

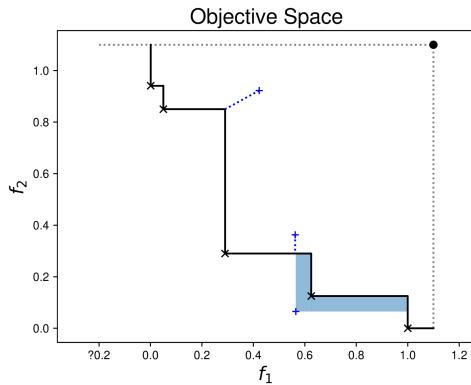


Figure 2: Illustration of the proposed UHVI fitness for three points in objective space (blue +). The top two points are dominated by the given set S of five points (black crosses) and are thus assigned a fitness of their distance to the empirical non-dominated set while the bottom point is non-dominated and thus gets assigned the hypervolume improvement with respect to S .

$\theta_{t+1} = G^f(\theta_t, U_{t+1})$ where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the single-objective function optimized by the optimizer and U_{t+1} encodes possible random variables sampled within one iteration if we consider a

randomized algorithm (and can be taken as constant in the case of a deterministic optimizer). The transition function G^f contains all updates done within the algorithm in one iteration.

We assume that in each iteration t , the optimizer returns a best estimate of the optimum, often called incumbent solution or recommendation. This is the solution that the optimizer would return if we stop it at iteration t . We denote this incumbent as $\mathcal{E} : \theta \in \Theta_n \mapsto \mathcal{E}(\theta) \in \mathbb{R}^n$ —mapping the state of the algorithm to the estimate of the optimum given this state.

The overall idea behind the subspace optimization and the Sofomore framework can then be formalized as in Algorithm 1: after initializing p single-objective algorithms with their states $\theta_1, \dots, \theta_p$ and denoting their transition functions as G_i^f ($1 \leq i \leq p$), we consider their incumbents or recommendations $\mathcal{E}(\theta_i)$ as the p search points that are expected to approximate the optimal p -distribution.

In each step of the Sofomore framework, we choose one of the algorithms (denoted by its number i , with $1 \leq i \leq p$) and run it τ_i iterations on the fitness $\Phi_{\text{UHVI}, X^{(-i)}}$ to update the recommendation $x^{(i)}$ while keeping all other recommendations fixed. It is important to note that the fitness used for algorithm i is actually changing *dynamically* with the optimization because it depends on all the other incumbents but $x^{(i)}$ which, over time, are expected to move towards the Pareto set as well.

Algorithm 1 proposes a generic framework where the order in which the single-objective algorithms are run and the number of

iterations for them are not explicitly defined. A simple strategy would be to choose the algorithms at random or in a given, fixed order and run each single-objective algorithm a fixed number of time steps. But also more elaborate strategies can be envisioned, for example based on the idea of multi-armed bandits [6]: we can log the changes in the fitness value of each incumbent over time and favor as the next chosen algorithms the ones that give the highest expected fitness improvements. Note also that the single-objective algorithms' types may be different such that we can combine local with global algorithms or even change the algorithms over time, allow restarts etc. In the following experimental validation of our concept, however, we choose a single optimization algorithm and a simple, random strategy to choose which of them to run next.

Algorithm 1 General Sofomore Framework, with fitness of (6)

```

1: Given: the initial states of  $p$  single-objective optimizers,
    $\theta_1, \dots, \theta_p$ 
2: Initialize incumbents:  $X = \{x^{(1)} = \mathcal{E}(\theta_1), \dots, x^{(p)} = \mathcal{E}(\theta_p)\}$ 
3: while not stopping criterion met do
4:   Choose  $i$  in  $\{1, \dots, p\}$  and  $\tau_i \in \mathbb{N}$ 
5:   REPEAT  $\tau_i$  times:
6:      $\theta_i \leftarrow G_i^{\Phi_{\text{UHVI}, X^{(-i)}}}(\theta_i, U_i)$            ▷ run  $i$ th algo on fitness
    $\Phi_{\text{UHVI}, X^{(-i)}}$ 
7:      $x^{(i)} \leftarrow \mathcal{E}(\theta_i)$                                 ▷ update  $x^{(i)}$  in  $X$ 
8:   end while
9: return  $x^{(1)}, \dots, x^{(p)}$ 

```

With a simple change, Algorithm 1 can be made parallelizable (resulting in slightly different search dynamics though): postponing the updates of the $x^{(i)}$ after every algorithm has been touched at least once makes the optimization of the fitness functions independent such that they can be performed in parallel.

Relation of Sofomore with other existing algorithms. We briefly discuss how some existing algorithms and algorithm frameworks relate to the new Sofomore proposal.

The coupling of single-objective algorithms to form a multiobjective one has been done before, especially in the MOEA/D framework [29]. In MOEA/D, p static search directions (in objective space) are defined via p (single-objective) scalarizing functions. Each of them is optimized in parallel with solutions potentially shared between neighboring search directions. On the contrary, the fitness in Sofomore is dynamic, depending on the other incumbents. Optimizing a set of scalarizing functions in classical approaches to multiobjective optimization have static optimization problems to solve without any interaction between them [21].

Many other EMO algorithms, such as NSGA-II, SMS-EMOA, or MO-CMA-ES are not covered by the Sofomore framework. One simple reason is that the UHVI is newly defined.

The already mentioned Newton algorithm on the hypervolume indicator fitness of [16] is probably the closest existing approach from Sofomore, but [16] needs to initialize the Newton algorithm with a set of non-dominated solutions in order for the algorithm to optimize due to the flat regions of its objective space. Also algorithms for expensive multiobjective optimization based on the optimization of the expected hypervolume improvement [25] can

be seen as related to Sofomore, although the proposal of new solutions in algorithms like SMS-EGO [22] or S-metric based ExI [10] use Gaussian Processes to model the objective function. These algorithms, in contrary to Sofomore, propose iteratively a single solution based on the *expected* hypervolume improvement over all known solutions and do not aim at replacing successively a single recommendation by another (better) one. Interesting to note is that algorithms like SMS-EGO and S-metric based ExI employ the expected hypervolume indicator improvement as fitness while the approach of Keane [18] “uses the Euclidean distance to the nearest vector in the Pareto front” [25].

4 COMO-CMA-ES

In this section, we instantiate Sofomore with the CMA-ES as single objective optimizer.

Regarding the choice of which optimization algorithm to run (and how long), we opt for a simple strategy: we sample a permutation from S_p , the set of all permutations on $\{1, \dots, p\}$ uniformly at random and use this fixed permutation to touch each algorithm i once in the order of the permutation. Once all algorithms have been touched, we then resample a new permutation. We run each algorithm for a single iteration. Letting the algorithms run for a too long period right from the start seems suboptimal. As the fitness is dynamic, we do not need to optimize it too precisely. We mainly have two requirements for the choice of single objective optimizers: (i) an optimization algorithm has to be stoppable at any iteration and resumable thereafter and (ii) an optimization algorithm needs to be able to give a good recommendation about the best estimate of the optimum, given its current state. The Covariance Matrix Adaptation Evolution Strategy (CMA-ES, [15]) is a natural choice. Not only is it a state-of-the-art algorithm for difficult blackbox optimization problems but also does it fulfill our requirements. In CMA-ES, the state of the algorithm θ is composed of a step size σ and the parameters of a multivariate normal distribution, namely a mean vector \mathbf{m} representing the favorite solution and a covariance matrix C . In addition, two n -dimensional evolution paths speed up step-size and covariance matrix adaptation. For each θ_i , the incumbent solution $x^{(i)} = \mathcal{E}(\theta_i)$ is the mean \mathbf{m} of the CMA algorithm. A convenient implementation of CMA-ES is via the ask and tell interface [7], where the ask function returns λ candidate solutions and the tell function updates the state from their fitness values. The interface allows to easily stop and resume the optimization and to integrate the dynamic fitness of Sofomore, see Algorithm 2. We call this instantiation of the Sofomore framework COMO-CMA-ES. The p CMA-ES instances are called kernels.

We see in particular how CMA-ES is integrated into Sofomore via its ask-and-tell interface. After choosing the next kernel i , the corresponding CMA-ES instance samples λ solutions (“ask”). It then evaluates them on the uncrowded hypervolume improvement based fitness defined in Eq (6)—given all other kernels being fixed. After sorting the λ solutions with respect to their fitness, COMO-CMA-ES feeds the sampled points with their fitness values back to the CMA-ES instance (“tell”) which updates all its internal algorithm parameters. Finally, the new mean of the corresponding CMA-ES instance updates the list of the COMO-CMA-ES’s proposed p

Algorithm 2 The COMO-CMA-ES: an instance of the Sofomore framework with the CMA-ES as single-objective optimizer

```

1: Required:
2:   objective function  $f = (f_1, \dots, f_k)$  in dimension  $n$ 
3:   lower and upper bounds for each variable  $\text{lower} \in \mathbb{R}^n$ 
   and  $\text{upper} \in \mathbb{R}^n$  of a region of interest
4:   number of desired solutions  $p$ 
5:   global initial step-size  $\sigma_0$  for all CMA-ES
6:   fixed reference point  $r$  for the hypervolume indicator

7: Initialization:
8:    $x^{(i)} = \text{uniformSample}(\text{lower}, \text{upper})$  for all  $1 \leq i \leq p$ 
9:   evaluate all  $x^{(i)}$  on  $f$  and store the  $f(x^{(i)})$  for later use
10:   $\text{es}^{(i)} \leftarrow (\mu/\mu, \lambda)$ -CMA-ES  $(x^{(i)}, \sigma_0)$  for all  $1 \leq i \leq p$ 

11: while not stopping criterion do
12:   sample uniformly at random a permutation  $\pi$  from all
      permutations on  $\{1, \dots, p\}$ 
13:   for  $i = 1$  to  $p$  do
14:      $\{s^1, \dots, s^\lambda\} \leftarrow \text{es}^{(\pi(i))}.ask()$      $\triangleright$  get  $\lambda$  offspring from
            $\triangleright \pi(i)$ th CMA-ES
15:     compute the fitness  $\Phi(s^j) = \Phi_{\text{UHVI}, X^{(\sim \pi(i))}}(s^j)$ 
       for all  $1 \leq j \leq \lambda$ 
16:      $\text{es}^{(\pi(i))}.tell(\Phi(s^1), \dots, \Phi(s^\lambda))$ 
17:      $x^{(\pi(i))} \leftarrow \text{es}^{(\pi(i))}.mean$ 
18:     update the stored objective vector  $f(x^{(\pi(i))})$ 
19:   end for
20: end while
21: Return  $x^{(1)}, \dots, x^{(p)}$ 

```

solutions. Note here that CMA-ES is usually not evaluating the mean of the sample distribution which therefore is done in line 18.

5 EXPERIMENTAL VALIDATION

We present in this section numerical experiments of the COMO-CMA-ES. Though, in principle, the algorithm can be defined for any number of k objectives, we present results only for $k = 2$. We use the pycma Python package [13] version 2.6.0 for CMA-ES as single-objective optimizer without further parameter tuning.

5.1 Test Functions and Performance Measures

For a matrix P and two vectors x and y , we denote

$$\text{Quad}(P, x, y) = (x - y)^\top P (x - y). \quad (7)$$

We also denote by $\mathbf{0}$ the all-zeros vector, $\mathbf{1}$ the all-ones vector, and e_k the unit vector with its only nonzero value at position k . Starting from a positive diagonal matrix Δ , and two independent orthogonal matrices O_1 and O_2 , we consider the classes of bi-objective convex quadratic problems **Sep- k** , **One** and **Two** defined as follows [23]

- $f_{1,\Delta}^{\text{sep-}k}(x) = \frac{\text{Quad}(\Delta, x, \mathbf{0})}{\text{Quad}(\Delta, \mathbf{0}, e_k)}, f_{2,\Delta}^{\text{sep-}k}(x) = \frac{\text{Quad}(\Delta, x, e_k)}{\text{Quad}(\Delta, \mathbf{0}, e_k)}$.
- $f_{1,\Delta}^{\text{one}}(x) = \frac{\text{Quad}(O_1^T \Delta O_1, x, \mathbf{0})}{\text{Quad}(O_1^T \Delta O_1, \mathbf{0}, \mathbf{1})}, f_{2,\Delta}^{\text{two}}(x) = \frac{\text{Quad}(O_2^T \Delta O_1, x, \mathbf{1})}{\text{Quad}(O_2^T \Delta O_1, \mathbf{0}, \mathbf{1})}$
- $f_{1,\Delta}^{\text{two}}(x) = \frac{\text{Quad}(O_1^T \Delta O_1, x, \mathbf{0})}{\alpha}, f_{2,\Delta}^{\text{two}}(x) = \frac{\text{Quad}(O_2^T \Delta O_2, x, \mathbf{1})}{\alpha}$
with $\alpha = \max(\text{Quad}(O_1^T \Delta O_1, \mathbf{0}, \mathbf{1}), \text{Quad}(O_2^T \Delta O_2, \mathbf{0}, \mathbf{1}))$.

If Δ is the identity matrix, we call the problems as **sphere-sep- k** in the first case and **bi-sphere** in the second and third cases (the rotations are ineffective). If $\Delta(i, i) = 10^{6 \frac{i-1}{n-1}}$ for $i = 1, \dots, n$, then we denote the problems as **elli-sep- k** , **elli-one** or **elli-two**. If $\Delta(1, 1) = 10^{-4}$, $\Delta(2, 2) = 10^4$ and $\Delta(i, i) = 1$ for $i = 3, \dots, n$, then we have **cigtab-sep- k** , **cigtab-one** or **cigtab-two**.

We fix the reference point to $r = (1, 1, 1)$. The scalings above ensure that the reference point is dominated by all Pareto fronts considered, and that the **Sep- k** and the **One** problems have the same Pareto front (see [23]) than the bi-sphere $g : [0, 1] \ni t \mapsto (1 - \sqrt{t})^2 \in \mathbb{R}$. Note that the expression does not depend on the dimension n .

We use two performance measurements in each run of an algorithm. First the **convergence gap** defined as the difference between an offset called hv_max and the hypervolume of the p points $\{x^{(1)}, \dots, x^{(p)}\}$ found by the algorithm (in case of COMO-CMA-ES or of the population for the other algorithms tested) called hv ; and second the **archive gap** defined as the difference between an offset called hvarchive_max and the hypervolume of all non-dominated points found by the algorithm called hvarchive . The setting of hv_max is done for each problem as the maximum hypervolume value of p kernels found so far anytime the problem was optimized in our machines, plus a small number ($< 10^{-14}$). For the **Sep- k** and the **One** problems, we take hvarchive_max as $1.1^2 - \int_0^1 g(t)dt = 1.21 - \frac{1}{6}$ which corresponds to the hypervolume of the theoretical Pareto front. For the two-class of problems, we use the analytic expression of their Pareto set [23] to sample a large number of points on the Pareto set, and compute their hypervolume as hvarchive_max . Thus for the **elli-two** problem in dimension 10, we sample 10^7 points.

5.2 Linear convergence of COMO-CMA-ES

We investigate the convergence of COMO-CMA-ES for different dimensions and number of kernels, and display the results on the **sphere-sep-1**, **elli-sep-1**, **cigtab-sep-1** and **elli-two** functions for $n = 10$ and $p = 31$. The global initial step-size is set to $\sqrt{10}$ and the initial lower, upper bounds (line 8 of Algorithm 2) respectively to $-5\mathbf{1}, 5\mathbf{1}$. In Figure 3, we observe linear convergence in the convergence gap (first column) on all test functions, starting roughly when all displayed ratios of non-dominated points reach 1 (second column). The last three columns of Figure 3 illustrate the eigen-spectra of the kernels covariance matrices. The first two columns reveal two phases.

First, the kernels incumbents approach the non-dominated region: for **sphere-sep-1** this takes about 1500 evaluations per kernel, for **elli-sep-1**, **cigtab-sep-1** and **elli-two** it takes about 5000, 4000 and 6600 evaluations per kernel. Afterwards, the convergence gap converges linearly. In our settings, there are 11 evaluations per kernel during the update of a kernel, thus for the $^{*-1}$ functions (which have the same Pareto set and front), the linear convergence rate is about $10^{-\frac{6 \times 11}{15000}}$ and for **elli-two**, it is about $10^{-\frac{3 \times 11}{5000}}$.

For the first 1000 function evaluations per kernel on **elli-sep-1**, there is no point dominating the reference point, which means that the algorithm started far from the Pareto front. Looking at **elli-two**, we confirm that it has a different Pareto front than the three other problems: $\text{hv_max} = 1.2099\dots$ (instead of $1.0327\dots$).

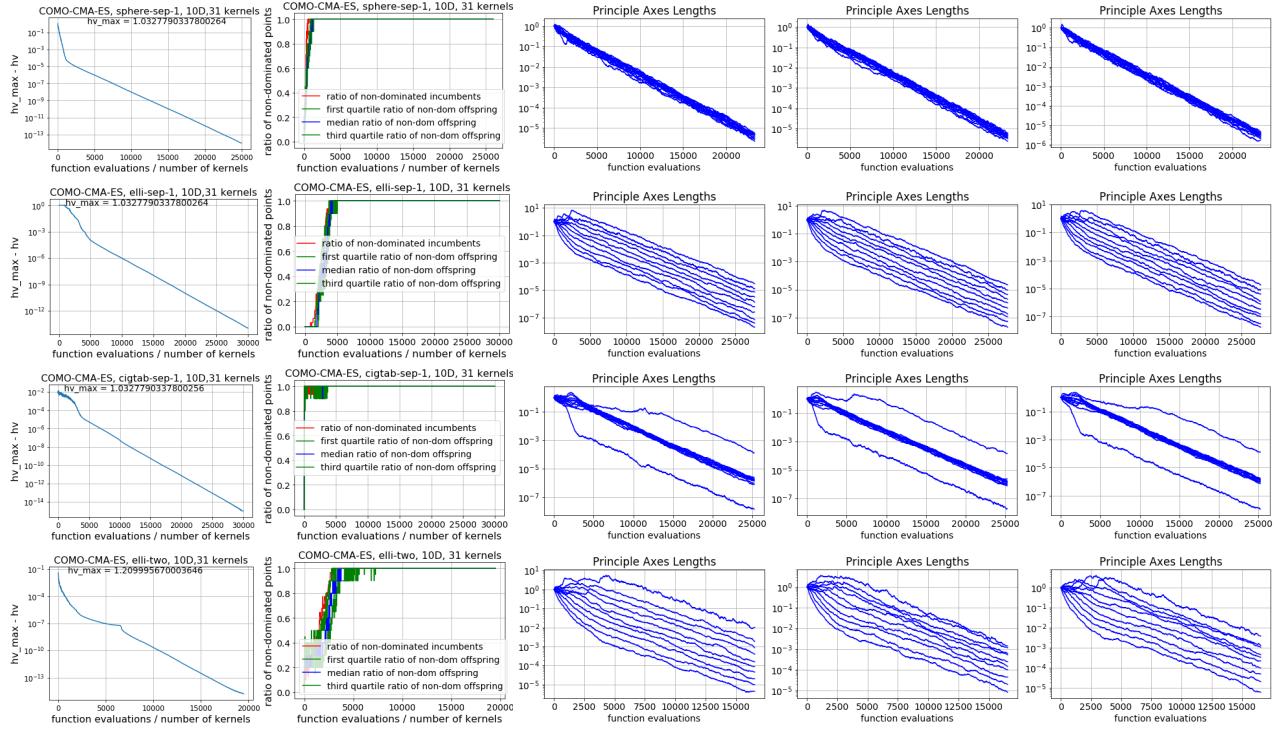


Figure 3: Convergence of COMO-CMA-ES on sphere-sep-1 (first row), ell-1sep-1 (second row), cigtab-sep-1 (third row) and ell-2 in 10D with 31 kernels. The first column represents the convergence gap. The second column is the ratio of non-dominated points among the kernels incumbents (red) and the quartiles of the 31 ratios of non-dominated points among each kernel's incumbent and its offspring (median in blue and the remaining quartiles in green). And the last three columns are square root eigenspectra of uniform randomly chosen 3 among the 31 kernels' covariance matrices.

The Uncrowded Hypervolume Improvement depends on other kernels' incumbents and therefore changes in each iteration. Yet, the last three columns are similar to what one would observe when optimizing a single objective convex-quadratic function with corresponding Hessian matrix. After a large enough number of iterations, the probability that the incumbents and their offspring are in the Pareto set becomes close to 1. Then if the incumbents $X = \{x^{(1)}, \dots, x^{(p)}\}$ is a subset of the Pareto set and x is non-dominated, Eq (6) becomes: $\Phi_{UHVI, X^{-i}}(x) = UHVI_r(x, X^{-i}) = HVI_r(x, X^{-i})$. Its Hessian on smooth bi-objective problems is $\nabla^2 \Phi_{UHVI, X^{-i}}(x) = (f_2(x) - f_2(x^{(i-1)})) \nabla^2 f_1(x) + (f_1(x) - f_1(x^{(i+1)})) \nabla^2 f_2(x) + \nabla f_2(x) \nabla f_1(x)^\top + \nabla f_1(x) \nabla f_2(x)^\top$. For our test functions, it is a linear combination of the single objectives Hessian matrices, up to a rank-one matrix and its transpose (the gradients are colinear on the Pareto set of bi-objective convex quadratic problems [23]). That might give a glimpse on the behaviour seen in the last three columns of Figure 3.

5.3 Comparing COMO-CMA-ES with MO-CMA-ES, NSGA-II and SMS-EMOA

We compare four multiobjective algorithms: COMO-CMA-ES, MO-CMA-ES [17], NSGA-II [8] and SMS-EMOA [4], by testing them on classes of bi-objective convex-quadratic problems. We draw

once and for all one rotation for **elli-one** in 10D and two different rotations for **elli-two** in 10D. The Simulated Binary Crossover operator (SBX) and the polynomial mutation are used for NSGA-II (run with the evoalgos package [27]) and SMS-EMOA (run with the Matlab version by Fabian Kretzschmar and Tobias Wagner [26]): we use a crossover probability of 0.7 and a mutation probability of 0.1, and the distribution indexes for crossover and mutation operators are both equal to 10. We use the version of MO-CMA-ES from [24]. The number of kernels for COMO-CMA-ES corresponds to the population size of the other algorithms, that we set to either 11 or 31, and the dimensions considered are 5 and 10. The global initial step-size of COMO-CMA-ES is set to 0.2 with initial lower, upper bounds (line 8 of Algorithm 2) set to the all-zeros and all-ones vectors. The initial population for the three other algorithms is sampled uniformly at random in $[0, 1]^n$.

We run each multiobjective optimization 4 times and display the convergence gap (of the population or the incumbent solutions of the kernels) and the archive gap.

In Figure 4, the values of the convergence gap reached by COMO-CMA-ES and MO-CMA-ES are several orders of magnitude lower than for the two other algorithms. On the 5-dimensional **bi-sphere**, COMO-CMA-ES and MO-CMA-ES appear to show linear convergence, where the latter appears to be about 30% faster than the former. On the **cigtab-sep-1** function, COMO-CMA-ES is initially

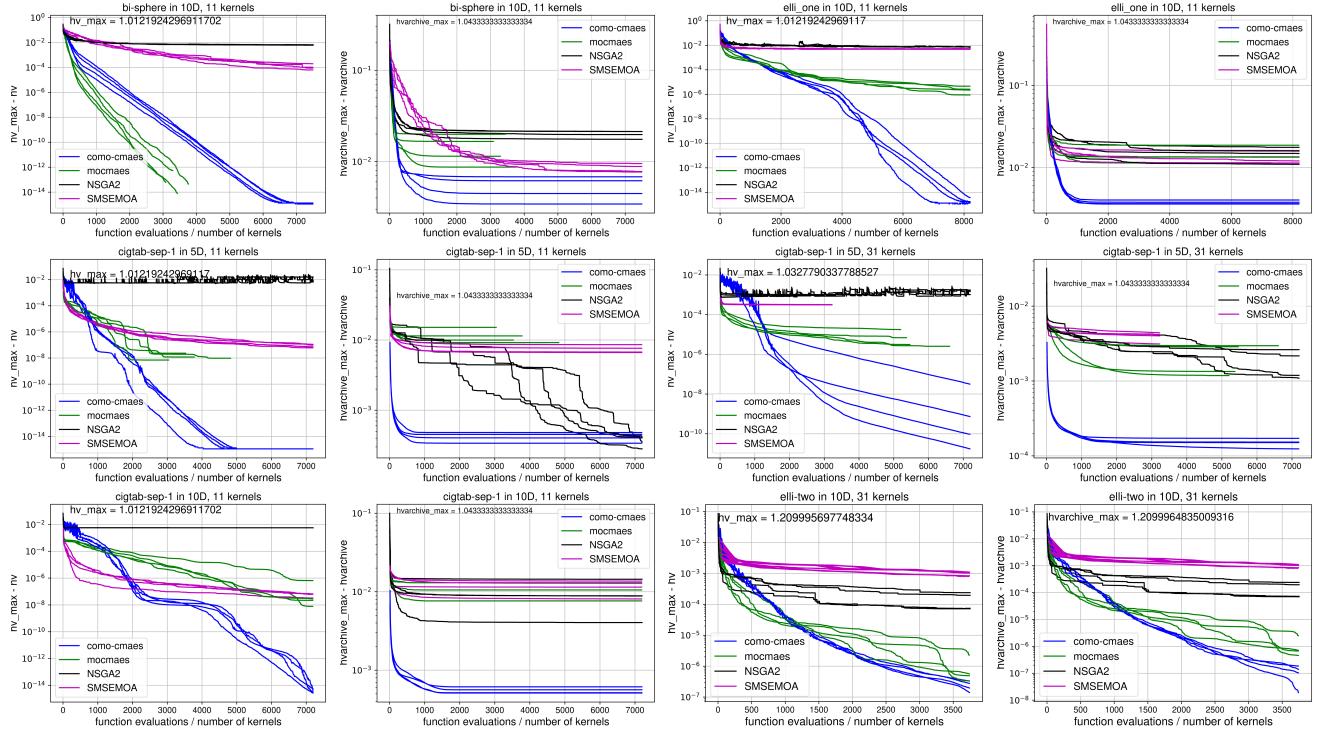


Figure 4: Convergence gaps (odd columns) and archive gaps (even columns) for bi-sphere, elli-one, cigtab-sep-1 and elli-two. Each algorithm is run 4 times, in 5D or 10D, with 11 or 31 kernels. The random matrices are drawn from the same seed in all the algorithms.

slow, but catches up after about 1000 evaluations per kernel. In all other cases, COMO-CMA-ES shows superior performance for the convergence gap. On the 10-dimensional **cigtab-sep-1**, COMO-CMA-ES shows a plateau between 2000 and 4000 evaluations per kernel. This kind of plateau cannot be observed in the MO-CMA-ES and the observed final convergence speed is better for COMO-CMA-ES than for MO-CMA-ES. The observed plateau is typical for the behavior of non-elitist multi-recombinative CMA-ES on the tablet function, because CSA barely reduces an initially large step-size before the tablet-shape has been adapted, which is related to the neutral subspace defect found in [20]. Elitism as in the MO-CMA-ES, on the other hand, also helps to decrease an initially too large step-size.

Although COMO-CMA-ES was not designed to perform well on the archive gap, it shows consistently the best results over all experiments. Only on the **cigtab-sep-1** in 5D with 11 kernels, NSGA-II reaches and slightly surpasses the archive gap of COMO-CMA-ES after 7000 function evaluations per kernel. This suggests, as expected from the known dependency between optimal step-size and population size [14], that COMO-CMA-ES adds valuable diversity while approaching the optimal p -distribution of the Pareto front at the same time.

6 CONCLUSIONS

We have proposed (i) the Sofomore framework to define multi-objective optimizers from single-objective ones, (ii) a fitness for

dominated solutions to be the distance to the empirical Pareto front (Uncrowded Hypervolume Improvement UHVI) and (iii) the non-elitist "comma" CMA-ES to instantiate the framework (COMO-CMA-ES). We observe that COMO-CMA-ES converges linearly towards the p -optimal distribution of the hypervolume indicator on several bi-objective convex quadratic problems. The COMO-CMA-ES appears to be robust to independently rotating the Hessian matrices of convex-quadratic problems, even if such rotations transform the Pareto set from a line segment to a bent curve. In our limited experiments, COMO-CMA-ES performed generally better than MO-CMA-ES, SMS-EMOA and the NSGA-II, w.r.t. convergence gap and archive gap while COMO-CMA-ES was solely designed to optimize the convergence gap. We conjecture that the advantage on the archive gap is due to (i) the large stationary variance obtained with non-elitist evolution strategies and (ii) the fitness assignment of dominated solutions which favors the vacant (uncrowded) space between non-dominated solutions and hence serves as implicit crowding distance penalty measure.

ACKNOWLEDGEMENTS

Part of this research has been conducted in the context of a research collaboration between Storengy and Inria. We particularly thank F. Huguet and A. Lange from Storengy for their strong support, practical ideas and expertise.

REFERENCES

- [1] 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, Orlando, Florida, USA, 87–102.
- [2] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. 2012. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theoretical Computer Science* 425 (2012), 75–103.
- [3] R. Berghammer, T. Friedrich, and F. Neumann. 2010. Set-based Multi-objective Optimization, Indicators, and Deteriorative Cycles. In *Genetic and Evolutionary Computation Conference (GECCO 2010)*. ACM, Portland, Oregon, 495–502. <https://doi.org/10.1145/1830483.1830574>
- [4] N. Beume, B. Naujoks, and M. Emmerich. 2007. SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. *European Journal of Operational Research* 181, 3 (2007), 1653–1669.
- [5] Karl Bringmann and Tobias Friedrich. 2011. Convergence of hypervolume-based archiving algorithms I: Effectiveness. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, Dublin, Ireland, 745–752.
- [6] Sébastien Bubeck and Nicolo Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5, 1 (2012), 1–122.
- [7] Yann Collette, Nikolas 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*, Rajan Filomeno Coelho and Piotr Breitkopf (Eds.). Wiley, New Jersey, 499–538. <https://hal.inria.fr/inria-00476172>
- [8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.
- [9] Michael Emmerich, Nicola Beume, and Boris Naujoks. 2005. An EMO algorithm using the hypervolume measure as selection criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, Guanajuato, Mexico, 62–76.
- [10] Michael Emmerich and Jan-willem Klinkenberg. 2008. *The computation of the expected improvement in dominated hypervolume of Pareto front approximations*. Technical Report 4-2008. Leiden Institute of Advanced Computer Science, LIACS.
- [11] D. E. Goldberg. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Massachusetts.
- [12] M. P. Hansen and A. Jaszkiewicz. 1998. *Evaluating The Quality of Approximations of the Non-Dominated Set*. Technical Report. Institute of Mathematical Modeling, Technical University of Denmark. IMM Technical Report IMM-REP-1998-7.
- [13] Nikolas Hansen, Youhei Akimoto, and Petr Baudis. 2019. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634. (Feb. 2019). <https://doi.org/10.5281/zenodo.2559634>
- [14] Nikolas Hansen, Dirk V Arnold, and Anne Auger. 2015. Evolution strategies. In *Springer handbook of computational intelligence*. Springer, Berlin, 871–898.
- [15] N. Hansen and A. Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9, 2 (2001), 159–195.
- [16] VAS Hernandez, O Schutze, H Wang, A Deutz, and M Emmerich. 2018. The Set-Based Hypervolume Newton Method for Bi-Objective Optimization. *IEEE transactions on cybernetics* in print (2018). (in print).
- [17] C. Igel, N. Hansen, and S. Roth. 2007. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation* 15, 1 (2007), 1–28.
- [18] Andy J. Keane. 2006. Statistical improvement criteria for use in multiobjective design optimization. *AIAA journal* 44, 4 (2006), 879–891.
- [19] J. Knowles, L. Thiele, and E. Zitzler. 2006. *A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers*. TIK Report 214. Computer Engineering and Networks Laboratory (TIK), ETH Zurich.
- [20] Oswin Krause, Tobias Glasmachers, and Christian Igel. 2017. Qualitative and quantitative assessment of step size adaptation rules. In *Proceedings of the 14th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*. ACM, Copenhagen, Denmark, 139–148.
- [21] K. Miettinen. 1999. *Nonlinear Multiobjective Optimization*. Kluwer, Boston, MA, USA.
- [22] Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. 2008. Multiobjective Optimization on a Limited Budget of Evaluations Using Model-Assisted S-Metric Selection. In *Parallel Problem Solving from Nature (PPSN 2008)*. Springer, Dortmund, Germany, 784–794.
- [23] Cheikh Toure, Anne Auger, Dimo Brockhoff, and Nikolaus Hansen. 2019. On Bi-Objective convex-quadratic problems. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, Lansing, Michigan, USA, 3–14.
- [24] T. Voß, N. Hansen, and C. Igel. 2010. Improved Step Size Adaptation for the MO-CMA-ES. In *Genetic and Evolutionary Computation Conference (GECCO 2010)*, J. Branke et al. (Eds.). ACM, Portland, OR, USA, 487–494.
- [25] Tobias Wagner, Michael Emmerich, André Deutz, and Wolfgang Ponweiser. 2010. On expected-improvement criteria for model-based multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*. Springer, Krakow, Poland, 718–727.
- [26] Tobias Wagner and Heike Trautmann. 2010. Online convergence detection for evolutionary multi-objective algorithms revisited. In *IEEE Congress on Evolutionary Computation*. IEEE, Barcelona, Spain, 1–8.
- [27] Simon Wessing. 2017. evoalgos: Modular evolutionary algorithms. Python package version 1. (2017). <https://pypi.python.org/pypi/evoalgos> [Online; accessed 31-January-2019].
- [28] Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. 2019. Multi-Objective Bayesian Global Optimization using expected hypervolume improvement gradient. *Swarm and evolutionary computation* 44 (2019), 945–956.
- [29] Q. Zhang and H. Li. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (2007), 712–731. <https://doi.org/10.1109/TEVC.2007.892759>
- [30] Eckart Zitzler and Simon Künzli. 2004. Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature*. Springer, Birmingham, UK, 832–842.
- [31] E. Zitzler and L. Thiele. 1998. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In *Conference on Parallel Problem Solving from Nature (PPSN V) (LNCS)*, Vol. 1498. Springer, Amsterdam, The Netherlands, 292–301.
- [32] Eckart Zitzler and Lothar Thiele. 1998. Multiobjective optimization using evolutionary algorithms - A comparative case study. In *International conference on parallel problem solving from nature*. Springer, Amsterdam, The Netherlands, 292–301.