Speeding Up Many-Objective Optimization
by Monte Carlo Approximations

Karl Bringmann[a], Tobias Friedrich[b,*], Christian Igel[c], Thomas Voß[d]

[a]Max-Planck-Institut für Informatik, Saarbrücken, Germany
[b]Friedrich-Schiller-Universität Jena, Germany
[c]Department of Computer Science, University of Copenhagen, Denmark
[d]Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany

Abstract

Many state-of-the-art evolutionary vector optimization algorithms compute the contributing hypervolume for ranking candidate solutions. However, with an increasing number of objectives, calculating the volumes becomes intractable. Therefore, although hypervolume-based algorithms are often the method of choice for bi-criteria optimization, they are regarded as not suitable for many-objective optimization. Recently, Monte Carlo methods have been derived and analyzed for approximating the contributing hypervolume. Turning theory into practice, we employ these results in the ranking procedure of the multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) as an example of a state-of-the-art method for vector optimization. It is empirically shown that the approximation does not impair the quality of the obtained solutions given a budget of objective function evaluations, while considerably reducing the computation time in the case of multiple objectives. These results are obtained on common benchmark functions as well as on two design optimization tasks. Thus, employing Monte Carlo approximations makes hypervolume-based algorithms applicable to many-objective optimization.

1. Introduction

Multi-objective optimization, also known as multi-criteria or vector optimization, is the basis of multiple criteria decision making [19, 22, 32, 33]. It is concerned with the optimization of vector-valued objective functions. The goal is to find or to approximate the set of Pareto-optimal solutions. A solution is Pareto-optimal if it cannot be improved in one objective without getting worse in another one. In recent years, it has become apparent that stochastic, population-based search algorithms such as evolutionary computing techniques are particularly well suited for solving vector optimization problems (e.g., see [15, 16]).

---

*Corresponding author: friedrich@uni-jena.de

Multi-objective evolutionary algorithms (MOEAs) have become broadly accepted methods in multi-criteria decision making and multiple criteria mathematical programming. It is known that the performance of MOEAs tends to deteriorate with an increasing number of objectives [14]. This is a general problem of vector optimization algorithms. For few objectives, MOEAs relying on the *contributing hypervolume* as the (second-level) sorting criterion are the methods of choice. These include the *evolution strategy with probabilistic mutation for multi-objective optimisation* (ESP, [26]), the *multi-objective covariance matrix adaptation evolution strategy* (MO-CMA-ES, [27, 28, 34]), the SMS-EMOA [4], and variants of the *indicator-based evolutionary algorithm* (IBEA, [41]). Despite the progress in developing algorithms for hypervolume computation (e.g., [3, 6, 7, 10, 21, 38, 40]), the computational complexity of calculating the contributing hypervolume prevents the broad application of these powerful MOEAs to objective functions with many (say, more than four) objectives.

Recently, several approximation algorithms for determining the least hypervolume contributor of a given Pareto-front approximation have been presented, for example, in [8, 11] or as part of the HypE MOEA [2]. Because of the good performance of HypE and encouraged by a preliminary study [36], we hypothesize that using such an approximation instead of the exact contributing hypervolume will make the aforementioned MOEAs applicable to problems with many objectives and that the resulting algorithms will push the boundaries of MOEAs for many-objective optimization.

In theory, the approximation allows for the application of hypervolume-based MOEAs to optimization problems with an arbitrary number of objectives. While there exist comparisons of approximation-based algorithms with other MOEAs [2], the effects of replacing the exact hypervolume calculation with an approximation algorithm on the overall performance of MOEAs have not been investigated in isolation. Apart from [36], there has been no empirical comparison of state-of-the-art MOEAs in which the exact hypervolume computation has been replaced by an approximation while fixing the other components of the algorithm.

Against this background, we employ the approximation within the steady-state MO-CMA-ES while all other components are kept fixed to empirically investigate whether the Monte Carlo approximation is actually useful in practice. In our experiments, using approximations indeed considerably reduced the computation time in the case of multiple objectives without impairing the quality of the obtained solutions.

The remainder of the document is structured as follows. The next section introduces the problem of determining the least hypervolume contributor. We briefly review results on Monte Carlo approximation of the least hypervolume contributor as well as the exact hypervolume algorithm. Then, we present our empirical evaluation before concluding with the results of our experiments and suggestions for future work.

2

2. Vector Optimization and the Least Hypervolume Contributor

We consider multi-objective optimization problems of the form

$$\boldsymbol{f}\colon X \to \mathbb{R}^m, \ \boldsymbol{f}(x) \mapsto (f_1(x), \dots, f_m(x)),$$

where X denotes the search space of the optimization problem and m refers to the number of objectives. Without loss of generality, we assume that all objectives are to be minimized. The number of objectives of all considered test problems is $m \geq 3$. *Pareto-dominance* is the fundamental concept for comparing candidate solutions of a multi-objective optimization problem. The candidate solution x' *weakly dominates* x and we write $x' \preceq x$ if

$$\forall i \in \{1, \dots, m\}\colon f_i(x') \leq f_i(x)$$

The solution x' *strictly dominates* x and we write $x' \prec x$ if *additionally*

$$\exists j \in \{1, \dots, m\}\colon f_j(x') < f_j(x)$$

hold.

Using the notion of dominance, the goal of multi-objective optimization can be defined as finding or approximating the set

$$X' = \{x \in X \mid \nexists x' \in X\colon x' \prec x\} \subseteq X,$$

which is called the *Pareto-optimal set*. The image of X' under $\boldsymbol{f}$ is referred to as the corresponding *Pareto-optimal front*.

The concept of dominance can be extended to sets. Let A and B be sets of candidate solutions. Then A weakly dominates B and we write $A \preceq B$ if every element in B is weakly dominated by at least one element in A.

2.1. Evolutionary Vector Optimization

Evolutionary algorithms (EAs, [20]) are iterative direct search heuristics that maintain a set of μ candidate solutions, the so-called parent population. In each iteration, λ new offspring solutions are generated. Then a new parent population is assembled from both the offspring and the former parent population. Candidate solutions with better objective function values are preferentially selected. In the elitist EA considered in this study, the parent population of the next generation is formed by the best μ of the new solutions and their parents. This requires sorting the solutions. However, the Pareto-dominance relation does not establish a total order. Therefore, incomparable candidate solutions need to be sorted by a so-called *second-level sorting criterion*. Given two incomparable individuals a and b (i.e. neither $a \preceq b$ nor $b \preceq a$ holds) and a Pareto front F, the second-level sorting criterion determines whether a or b is more valuable in the context of F (cf. [41]). The contributing hypervolume is one of the most popular second-level sorting criteria due to its attractive theoretical properties, and it is deployed in most recent multi-objective evolutionary algorithms.

2.2. The Contributing Hypervolume

The hypervolume measure or $\mathcal{S}$-metric (see [42]) of a population A is the volume of the union of regions of the objective space which are dominated by A and bounded by some appropriately chosen reference point $r \in \mathbb{R}^m$, that is,

$$\text{HYP}(A) := \text{VOL}\left(\bigcup_{a \in A} \left[f_1(a), r_1\right] \times \cdots \times \left[f_m(a), r_m\right]\right),$$

with $\text{VOL}(\cdot)$ being the Lebesgue measure. One of the unique features of the hypervolume indicator is that it is, up to weighting objectives, the only known indicator which is strictly Pareto compliant [43], that is, given two sets A and B the indicator values A higher than B if A dominates B. It has further been shown by Bringmann and Friedrich [13] that the worst-case approximation factor of all possible Pareto fronts obtained by any hypervolume-optimal set of fixed size μ is asymptotically equal to the best worst-case approximation factor achievable by any set of size μ, namely $\Theta(1/\mu)$ for additive approximation and $1 + \Theta(1/\mu)$ for relative approximation. The authors have shown in [23] that by considering a transformed variant of the hypervolume indicator, the *logarithmic hypervolume indicator*, a close-to-optimal multiplicative approximation ratio can be achieved. For these reasons, the hypervolume indicator is a popular second-level sorting criterion in many recent multi-objective evolutionary algorithms (MOEAs).

When using the hypervolume as a second-level sorting criterion for comparing incomparable individuals, we measure the respective contribution of each individual to the total hypervolume. The contributing hypervolume of an individual $a \in A$ is given by

$$\text{CON}(a, A) := \text{HYP}(A) - \text{HYP}(A \setminus \{a\}).$$

Note that the contributing hypervolume of a dominated individual is zero. Thus, in the following we assume that all dominated individuals have been removed from A before contributing hypervolumes are computed or estimated. The contribution $\text{CON}(a, A)$ is an important measure since instead of using the hypervolume directly, most hypervolume based algorithms such as the steady-state MO-CMA-ES or the SMS-EMOA remove, in the selection step, the individual

$$a_1 := \operatorname*{argmin}_{a \in A} \left(\text{CON}(a, A)\right)$$

contributing the least hypervolume to the population A.[1]

---

[1]To be precise, the algorithms consider not the whole population A but the subset A' of individuals having the worst rank w.r.t. non-dominated sorting [17]. This set is constructed iteratively starting from $A' = A$ as follows. If A' contains dominated individuals, all non-dominated individuals are removed from A'. This removal process is iterated until the remaining set A' does not contain dominated individuals, and the individual contributing least

3. Computing the Contributing Hypervolume

In this section, we summarize an exact algorithm and two approximation schemes for calculating the contributing hypervolume.

3.1. Exact Computation

In order to determine a_1, the usual way is calculating $\mathrm{HYP}(A)$ and $\mathrm{HYP}(A \setminus \{a\})$ for all $a \in A$. This can be done by $(|A| + 1)$ hypervolume calculations with one of the many available hypervolume algorithms. Unfortunately, as the problem is #P-hard [9], none of them can run in time polynomial in m unless P = NP. In fact, assuming the widely believed exponential time hypothesis [30], the runtime of all algorithms computing the hypervolume must be $|A|^{\Omega(m)}$ [12]. Note that this only holds in the worst-case. The average-case complexity can be polynomial in the number of objectives [12].

Many algorithms have been present recently for calculating the hypervolume (e.g. [3, 6, 7, 10, 21, 38, 40]). We use the algorithm of Bringmann and Friedrich [10] which computes all contributions $\mathrm{CON}(a, A)$, $a \in A$, in only one pass. This saves a factor of $|A|$ compared to most other hypervolume algorithms and gives a total runtime of $\mathcal{O}(|A|^{m/2} \log(|A|))$ to compute all $|A|$ hypervolume contributions. For dimension $m = 3$ there is an even faster algorithm by Emmerich and Fonseca [21], which computes all hypervolume contributions in time $\mathcal{O}(|A| \log(|A|))$. As we are more interested in higher-dimensional problems ($m \geq 5$), we use the algorithm of Bringmann and Friedrich [10] for all dimensions.

3.2. Probably Approximately Correct Approximation

In our experiments we want to compare this exact calculation of the hypervolume with an approximation algorithm. It is known that the hypervolume can be approximated very efficiently by an FPRAS (fully polynomial-time randomized approximation scheme) [9]. Unfortunately, an approximation of the hypervolume does not yield an approximation of CON. Even worse, $\mathrm{CON}(a, A)$ is not only #P-hard to calculate exactly, it is also NP-hard to approximate by a factor of $2^{m^{1-\varepsilon}}$ for all $\varepsilon > 0$ [8, 11]. Though CON is therefore not approximable in time polynomial in $|A|$ and m, there are still a few approximation algorithms for CON (see [2, 8, 11]). The approximation algorithms presented in [2, 8, 11] are Monte Carlo algorithms based on different sampling techniques. The algorithm HypE [2] is a MOEA using hypervolume estimations at a user-specified confidence level to guide the search. However, here we want to compare a standard MOEA with and without hypervolume approximation to study the effects of the approximation. For this, we use the approximation algorithm of Bringmann and Friedrich [8, 11]. We now briefly describe this approach, which we

---

hypervolume to A' is removed from the population A.

will later improve in Section 3.3. This Monte Carlo algorithm returns, for a population A and arbitrary small $\varepsilon, \delta > 0$, an individual $\widetilde{a_1}$ with the property

$$\Pr \left[\mathrm{CON}(\widetilde{a_1}, A) \leq (1 + \varepsilon) \, \mathrm{CON}(a_1, A) \right] \geq 1 - \delta. \tag{1}$$

The algorithm samples in the minimal bounding boxes of all contributions and conducts a race between the different candidates until an individual $\widetilde{a_1}$ is found which (with high probability) has a contribution very close to the hypervolume contribution of a_1.

The runtime of this algorithm is bounded by $\mathcal{O}(m \, |A| \, (|A| + H))$, where H is a measure of hardness of the instance. It is polynomial in m and $|A|$ for most practical instances, but unbounded in the worst case. More precisely, H is defined as

$$
\begin{aligned}
H := &\left(\frac{\mathrm{BB}(a_1, A)}{\mathrm{CON}(a_2, A) - \mathrm{CON}(a_1, A)} \right)^2 \\
&+ \sum_{a_1 \neq a \in A} \left(\frac{\mathrm{BB}(a, A)}{\mathrm{CON}(a, A) - \mathrm{CON}(a_1, A)} \right)^2,
\end{aligned}
\tag{2}
$$

where a_1 denotes the individual with the smallest contribution, a_2 denotes the individual with the second smallest contribution, and $\mathrm{BB}(a, A)$ denotes the volume of the smallest bounding box of the contribution $\mathrm{CON}(a, A)$. By definition, H is unbounded and can even be undefined if there is no unique least contributor. However, in such cases an abortion criterion bounds the runtime. In general, H is small if $\mathrm{BB}(a, A) \approx \mathrm{CON}(a, A)$ and $\mathrm{CON}(a_1, A) \ll \mathrm{CON}(a, A)$ for all $a \in A \backslash \{a_1\}$. On the other hand, H is large if either (i) there is an individual with a large bounding box $\mathrm{BB}(a, A)$ but a small contribution $\mathrm{CON}(a, A)$, or (ii) there are two or more boxes contributing the minimal contribution or only slightly more than it, that is, for all $\mathrm{CON}(a, A) - \mathrm{CON}(a_1, A)$, $a \neq a_1$, is very small.

3.3. Fast Approximate Computation

As has been pointed out, approximating the least hypervolume contributor is an NP-hard problem [8, 11]. For fixed error bounds and error probabilities, the above described approximation scheme can degenerate to an exponential runtime. For guiding the search in a randomized search heuristic this seems inappropriate. Despite the fact that the approximation algorithm is reported to have a very fast empirical average-case performance [8, 11], we observed that difficult situations do indeed occur for typical benchmark problems, and sometimes very many samples are needed to achieve the specified error bound and error probability. These slow instances have a very large hardness value H (see Equation (2)), for example, because the contribution is extremely small compared to the bounding box and most samples do not lie in the contribution. This is unavoidable for a probably approximately correct approximation, but undesirable for a practical optimization algorithm.

6

To address these situations, we propose a heuristic that stops the overall selection process whenever a certain threshold of total samples has been reached. On early stopping, the current estimate of the least contributor is considered further by the selection scheme. Note that, in contrast to the approximation scheme described in Section 3.2, this algorithm is not anymore probably approximately correct with parameters ε and δ (cf. Equation (1)), but it results in a very fast algorithm which still gives competitive results with respect to the quality of the final Pareto-front approximation.

This new approximation scheme comes with an additional threshold parameter. In order to determine a good value for this parameter, we conducted a preliminary study of the threshold parameter for the DTLZ benchmark set [18]. As a testbed, we chose the objective function DTLZ 2 with eight objectives and considered a maximum number of 10^3, 10^4, 10^5 and 10^6 samples for the hypervolume approximation scheme. We conducted 25 independent trials for every parameter setting and analyzed the quality of the resulting Pareto-front approximations in terms of the absolute hypervolume indicator. Moreover, we recorded the required running time for every parameter setup.

The results of the parameter study are presented in Figure 1. Limiting the number of samples directly affects the running time of the selection scheme, and thus, of the overall algorithm. For a threshold of 10^3 and 10^4 samples, respectively, the quality of the resulting Pareto-front approximations is negatively affected and the absolute hypervolume starts to fluctuate as it nears the Pareto-optimal front (see Figure 1, bottom). In case of higher thresholds (10^5 and 10^6 samples), the quality remains stable and is on par with the quality obtained when considering the approximation scheme without a sample threshold. We therefore use sample threshold 10^5 for our fast approximation algorithm.

4. Empirical Evaluation

We compared the two approximated hypervolume indicators to the exact hypervolume indicator w.r.t. the influence on the performance (in terms of the quality of the final Pareto-front approximation) and the running time of MOEAs. To this end, we deploy all three indicators in the steady-state variant of the multi-objective covariance matrix evolution strategy (MO-CMA-ES, [27, 34, 37]). However, we expect the results to carry over to any MOEA also relying on the hypervolume indicator as second-level sorting criterion.

4.1. MO-CMA-ES

The MO-CMA-ES relies on the Pareto-dominance relation and a second-level sorting criterion for selection, which is state-of-the-art in MOEAs since [17]. The algorithm builds on the principles of the single-objective covariance matrix adaptation strategy (CMA-ES, [24, 31, 34]), which is a variable metric algorithm adapting the shape and strength of its Gaussian search distribution. The claim that the "CMA-ESs represent the state-of-the-art in evolutionary optimization in real-valued $\mathbb{R}^n$ search spaces" [5] is backed up by many performance comparisons across different suites of benchmark problems (e.g., see the competition
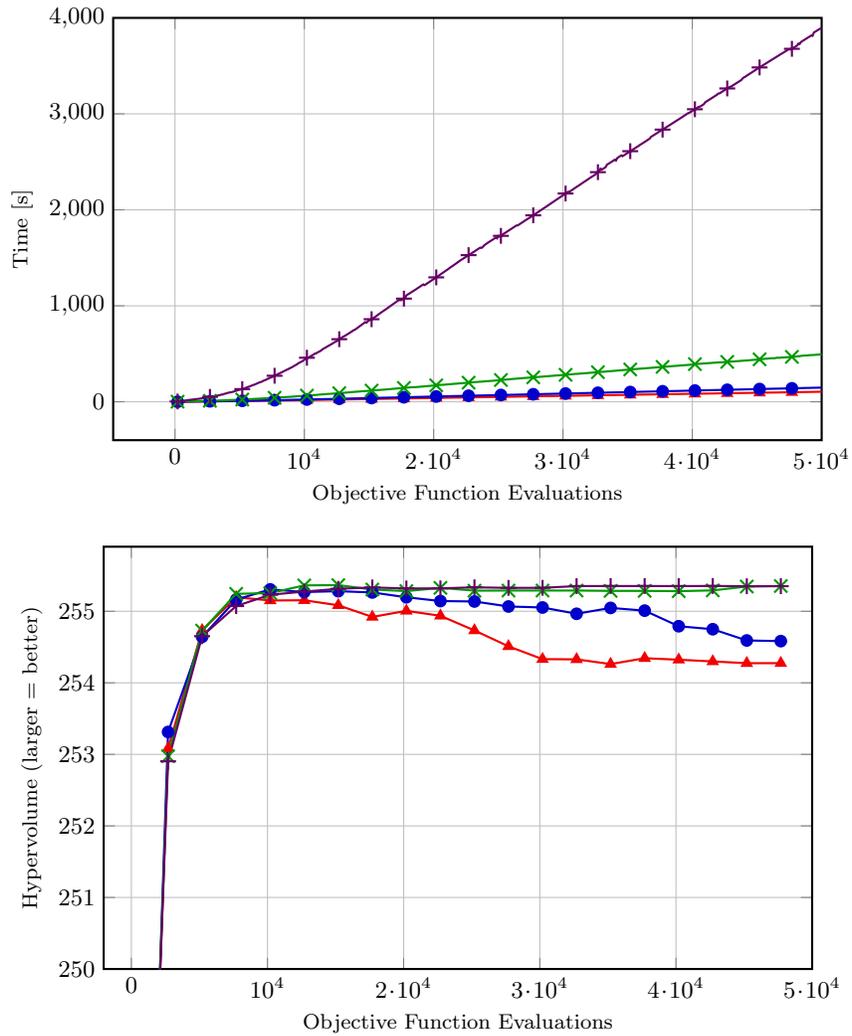
Figure 1: Comparison of different sample thresholds (10^3: ▲, 10^4: ●, 10^5: ✕, 10^6: ✛) for the benchmark function DTLZ 2 with eight objectives. Both the quality of the Pareto-front approximations (in terms of the absolute hypervolume) and the runtime requirements for the different parameter settings are shown as the median of 25 independent trials.

results in [1, 25]). Here, we use the most recent variant of the $(\mu + 1)$-MO-CMA-ES presented in [37]. For empirical evaluations of the MO-CMA-ES see, for example, [27, 28, 37].

4.2. Experimental Setup

We compared the MO-CMA-ES using an approximation of the least hypervolume contributor to the results of the original MO-CMA-ES with exact hypervolume computation. More precisely, we compare the following indicators:

- Exact computation as described in Section 3.1.

- Probably approximately correct as described in Section 3.2.

- Fast approximation with sample threshold 10^5 as described in Section 3.3.

This allows us to isolate the influence of the indicator by altering only the environmental selection procedure. We applied the algorithms to several classes of benchmark functions that are scalable to an arbitrary number of objectives m. We considered the seven constrained functions DTLZ 1–7 [18] with search space dimension 30. The number of objectives was chosen to be 3, 5, and 7. In addition, we considered two real-world many-objective optimization problems. The first one deals with the design of 2D airfoil shapes that are encoded by 10 real-valued parameters, the so-called PARSEC 10 parameter set. An airfoil shape is then evaluated with respect to six objectives by means of a computational fluid dynamics simulation (see [39]). The second one considers the optimization of centrifugal pump designs with respect to eight objectives (see [35]).

For all experiments, the number of parent individuals was set to $\mu = 50$. We conducted 25 independent trials with 50,000 objective function evaluations each. For both approximation algorithms, we used the parameters $\varepsilon = 10^{-2}$ and $\delta = 10^{-2}$ (cf. Equation (1)). The experiments were carried out on a cluster of technically equivalent workstations with 2.93 GHz Intel Quad-Core Xeon processors running Linux. We used the GNU compiler chain and enabled compiler optimizations according to the following command line arguments: `-O3 -ffast-math -msse4 -mtune=core2`. The reported runtimes refer to the overall CPU times.

We monitored the performance of the algorithms after every 5,000th objective function evaluation and carried out the statistical evaluation after 25,000 and 50,000 function evaluations. We relied on the hypervolume indicator to compare the Pareto-front approximations obtained by the three optimizers. In case of the benchmark functions DTLZ 1–7 and the centrifugal pump design problem, the reference point was determined from the union of all Pareto-front approximations. In case of the airfoil shape optimization problem, we chose the reference point $r = (0.00516, 0.00606, 0.00982, 0.30806, 0.92314, 0.65460)$ as suggested in [39]. We applied the statistical evaluation procedure described in [37] to evaluate our experiments employing the Wilcoxon rank-sum test to verify statistical significance. All experiments were implemented using the Shark machine learning library [29], which is available online.
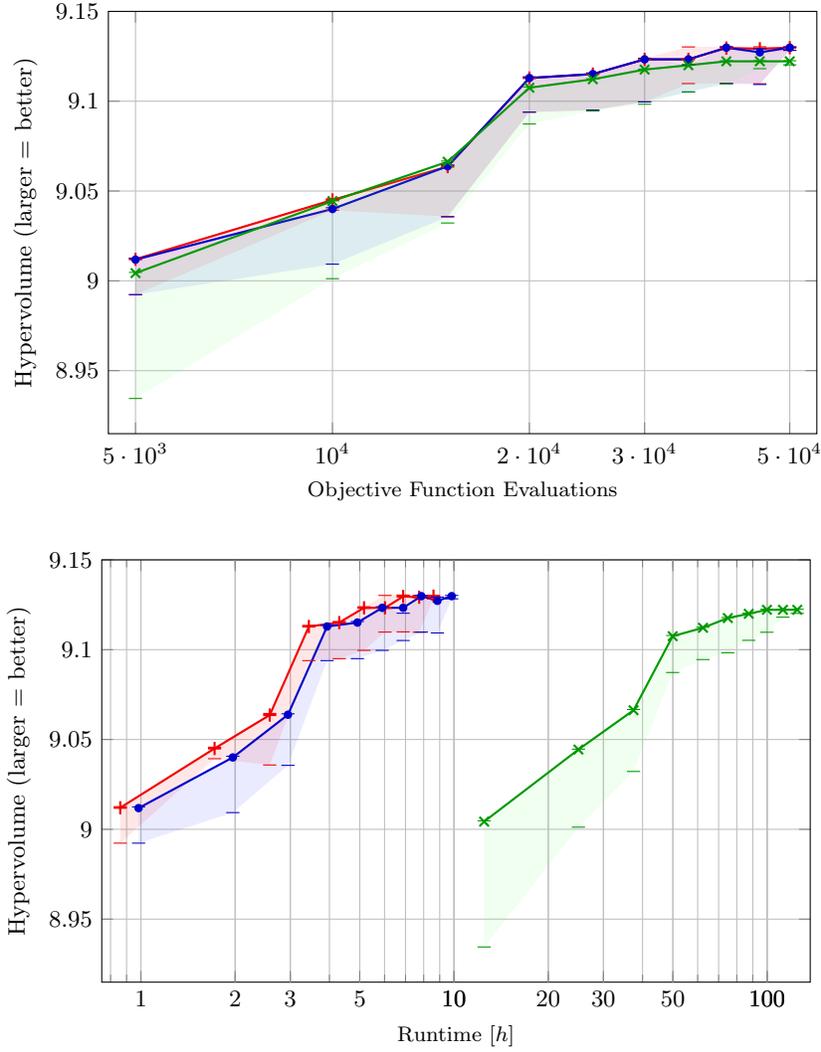
9

Figure 2: Achieved absolute hypervolume (the higher the better) for the objective function DTLZ 3 with seven objectives for the MO-CMA-ES with exact indicator (—✕—), with approximate indicator (—●—) and with its heuristic variant with sample threshold 10^5 (—+—). Shown are the medians and the corresponding lower and upper quartiles. The areas between both quartiles are shaded. Note that the 75 % percentile and median are very close. In the upper plot, where the x-axis is the number of fitness function evaluations, the two algorithms behave similarly. After roughly 25,000 fitness evaluations they are very close to the Pareto front. The lower plot shows the hypervolume depending on the actual running time on a logarithmic scale, demonstrating the performance gain by using the Monte Carlo approximations. Plots for all other objective functions with five or more objectives look similar.

	Exact computation		Probably approximately correct approximation		Fast approximation with sample threshold 10^5	
DTLZ 1 (3 objectives)		7m	16m	(2x slower)	16m	(2x slower)
DTLZ 1 (5 objectives)	4h	2m	24m	(10x faster)	24m	(10x faster)
DTLZ 1 (7 objectives)	3d 19h	12m	21h 21m	(4.3x faster)	13h 31m	(6.7x faster)
DTLZ 2 (3 objectives)		14m	1m	(14x faster)	1m	(14x faster)
DTLZ 2 (5 objectives)	18h	6m	31m	(35x faster)	31m	(35x faster)
DTLZ 2 (7 objectives)	23d 0h	43m	1d 11h 44m	(16x faster)	17h 53m	(31x faster)
DTLZ 3 (3 objectives)		3m	1m	(3x faster)	1m	(3x faster)
DTLZ 3 (5 objectives)		54m	21m	(2.6x faster)	21m	(2.6x faster)
DTLZ 3 (7 objectives)	5d 4h	48m	9h 50m	(13x faster)	8h 36m	(15x faster)
DTLZ 4 (3 objectives)		8m	1m	(8x faster)	1m	(8x faster)
DTLZ 4 (5 objectives)	2h	20m	41m	(3.4x faster)	41m	(3.4x faster)
DTLZ 4 (7 objectives)	3d 16h	48m	2d 3h 4m	(1.7x faster)	2h 3m	(43x faster)
DTLZ 5 (3 objectives)		32m	1m	(32x faster)	1m	(32x faster)
DTLZ 5 (5 objectives)	5h	14m	34m	(9.2x faster)	34m	(9.2x faster)
DTLZ 5 (7 objectives)	9d 3h	24m	20h 18m	(11x faster)	29m	(454x faster)
DTLZ 6 (3 objectives)	15h	46m	1m	(946x faster)	1m	(946x faster)
DTLZ 6 (5 objectives)	6d 19h	40m	43m	(228x faster)	43m	(228x faster)
DTLZ 6 (7 objectives)	41d 18h	14m	1d 3h 1m	(37x faster)	23m	(2614x faster)
DTLZ 7 (3 objectives)		4m	1m	(4x faster)	1m	(4x faster)
DTLZ 7 (5 objectives)	2h	44m	36m	(4.6x faster)	35m	(4.7x faster)
DTLZ 7 (7 objectives)	8d 21h	7m	1d 5h 23m	(7.3x faster)	(10h 7m)	(21x faster)
Airfoil (6 objectives)	6d 12h	57m	1d 5h 45m	(5.3x faster)	23h 34m	(6.7x faster)
Pump (8 objectives)	>100d		8d 2h 13m	(>12x faster)	4d 19h 27m	(>21x faster)

Table 1: Experimental results for DTLZ benchmark set obtained by the MO-CMA-ES with exact and two approximate calculations of the hypervolume indicator after 50,000 objective function evaluations. The differences in achieved hypervolumes were not statistically significant according to a Wilcoxon rank-sum test at a confidence level of $\alpha = 0.05$ for all test functions except DTLZ 7 with 7 objectives. The median running time that each of the algorithms required to complete one independent trial is listed and the average overall speedup is summarized.

4.3. Results

Table 1 summarizes our experimental results. In terms of the achieved hypervolume, the MO-CMA-ES relying on an approximation of the least hypervolume contributor performed at least on par with the variant employing the exact hypervolume indicator across the set of benchmark functions. More precisely, the differences in achieved hypervolumes were not statistically significant according to a Wilcoxon rank-sum test at a confidence level of $\alpha = 0.05$ for all test functions except DTLZ 7 with 7 objectives. The final objective values are therefore omitted in Table 1.

Table 1 shows clearly that the MO-CMA-ES achieves an enormous speed-up if the indicator is only approximated, especially for more than 4 dimensions. The speedup is the largest for the heuristically improved approximation introduced in Section 3.3 of this paper. However, we also observe that for very complicated test functions such as the disconnected fronts of DTLZ 7 in 7 dimensions, a sample threshold of 10^5 might not suffice. Here, we observe a tradeoff between speed (sample threshold 10^5) and quality (probably approximately correct com-

putation).

Figure 2 illustrates the typical behavior of the algorithms as a function of time and number of objective function evaluations. Shown are the medians over the 25 trials and the corresponding lower and upper quartiles (i.e., 25th and 75th percentiles). As expected, the hypervolume as a function of the number of objective function evaluations behaves similarly. On the other hand, both variants with approximated hypervolume finished all 50,000 objective functions evaluations in less than 10 hours, while the exact version needed more than 10 hours for only the first 5000 objective function evaluations.

A closer look at Figure 2 reveals that the algorithms using the Monte Carlo approximation perform even a bit better. The final medians are slightly higher, and, more prominently, the lower quartiles are larger. That is, there are fewer trials reaching only low hypervolume values within the given budget. A possible explanation might be that the approximation adds a little noise to the otherwise deterministic selection operator, which turns out to be beneficial for the evolutionary process.

5. Conclusions

We empirically investigated the effects of replacing the exact hypervolume indicator with two different Monte Carlo approximations on the performance of multi-objective evolutionary algorithms (MOEAs). We evaluated whether a state-of-the-art MOEA relying on hypervolume-indicator-based selection is affected by the potential errors made by approximating the least hypervolume contributor. The results show that the performance of the algorithms in terms of the quality of the Pareto front approximation given a budget of objective function evaluations does not suffer from the additional noise introduced by the Monte Carlo approximation. In some trials, the approximation, which introduces noise into the otherwise deterministic, greedy indicator-based selection scheme, led to better performance. We observed a vast reduction of the running time even for few objectives, when relying on the approximation scheme. In general, the higher the number of objectives the more pronounced the performance advantage of using Monte Carlo approximation becomes. Hence, by employing Monte Carlo approximations, hypervolume-based MOEAs become applicable to many-objective optimization.

References

[1] A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1777–1784. IEEE Press, 2005.

[2] J. Bader and E. Zitzler. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011.

[3] N. Beume. S-metric calculation by considering dominated hypervolume as Klee's measure problem. *Evolutionary Computation*, 17(4):477–492, 2009.

[4] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[5] H.-G. Beyer. Evolution strategies. *Scholarpedia*, 2(8):1965, 2007.

[6] L. Bradstreet, L. While, and L. Barone. A fast incremental hypervolume algorithm. *IEEE Transactions on Evolutionary Computation*, 12(6):714–723, 2008.

[7] K. Bringmann. An improved algorithm for Klee's measure problem on fat boxes. *Computational Geometry: Theory and Applications*, 45(5-6):225–233, 2012.

[8] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. In *Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, volume 5467 of *LNCS*, pages 6–20. Springer, 2009.

[9] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. *Computational Geometry: Theory and Applications*, 43:601–610, 2010.

[10] K. Bringmann and T. Friedrich. An efficient algorithm for computing hypervolume contributions. *Evolutionary Computation*, 18(3):383–402, 2010.

[11] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. *Theoretical Computer Science*, 425:104–116, 2012.

[12] K. Bringmann and T. Friedrich. Parameterized average-case complexity of the hypervolume indicator. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO)*, pages 575–582. ACM Press, 2013.

[13] K. Bringmann and T. Friedrich. Approximation quality of the hypervolume indicator. *Artificial Intelligence*, 195:265–290, 2013.

[14] D. Brockhoff. Theoretical aspects of evolutionary multiobjective optimization. In A. Auger and B. Doerr, editors, *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, pages 101–139. World Scientific Publishing, 2011.

[15] C. Coello Coello, G. B. Lamont, and D. A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2 edition, 2007.

[16] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, 2001.

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[18] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, pages 825–830. IEEE Press, 2002.

[19] M. Ehrgott. *Multicriteria Optimization*. Springer, 2 edition, 2010.

[20] A. E. Eiben and J. E. Smith. *Introduction to evolutionary computing*. Natural Computing. Springer, 2008.

[21] M. T. M. Emmerich and C. M. Fonseca. Computing hypervolume contributions in low dimensions: Asymptotically optimal algorithm and complexity results. In *Proceedings of the Evolutionary Multi-Criterion Optimization (EMO)*, volume 6576 of *LNCS*, pages 121–135. Springer, 2011.

[22] J. Figueira, S. Greco, and M. Ehrgott, editors. *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research & Management Science*. Springer, 2004.

[23] T. Friedrich, K. Bringmann, T. Voß, and C. Igel. The logarithmic hypervolume indicator. In H. Beyer and W. B. Langdon, editors, *Proceedings of the 11th International Workshop on Foundations of Genetic Algorithms (FOGA)*, pages 81–92. ACM Press, 2011.

[24] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[25] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO)*, pages 1689–1696. ACM, 2010.

[26] S. Huband, P. Hingston, L. While, and L. Barone. An evolution strategy with probabilistic mutation for multi-objective optimisation. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, volume 4, pages 2284–2291, 2003.

[27] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

[28] C. Igel, T. Suttorp, and N. Hansen. Steady-state selection and efficient covariance matrix update in the multi-objective CMA-ES. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, volume 4403 of *LNCS*, pages 171–185. Springer, 2007.

[29] C. Igel, T. Glasmachers, and V. Heidrich-Meisner. Shark. *Journal of Machine Learning Research*, 9:993–996, 2008.

[30] R. Impagliazzo and R. Paturi. The complexity of k-SAT. In *Proceedings of the 14th IEEE Conference on Computational Complexity (CCC)*, pages 237–240, 1999.

[31] S. Kern, S. Müller, N. Hansen, D. Büche, J. Ocenasek, and P. Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms – A comparative review. *Natural Computing*, 3:77–112, 2004.

[32] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *Kluwer's International Series in Operations Research & Management Science*. Kluwer Academic Publishers, 1999.

[33] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*, volume 176 of *Mathematics in Science and Engineering*. Academic Press, 1985.

[34] T. Suttorp, N. Hansen, and C. Igel. Efficient covariance matrix update for variable metric evolution strategies. *Machine Learning*, 75(2):167–197, 2009.

[35] R. K. Ursem. Centrifugal pump design: Three benchmark problems for many-objective optimization. Technical Report 2010-01, Grundfos Management A/S, 2010.

[36] T. Voß, T. Friedrich, K. Bringmann, and C. Igel. Scaling up indicator-based MOEAs by approximating the least hypervolume contributor: A preliminary study. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO): Workshop on Theoretical Aspects of Evolutionary Multiobjective Optimization*, pages 1975–1978. ACM Press, 2010.

[37] T. Voss, N. Hansen, and C. Igel. Improved step size adaptation for the MO-CMA-ES. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation Conference (GECCO)*, pages 487–494. ACM Press, 2010.

[38] L. While, L. Bradstreet, and L. Barone. A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation,*, 16(1): 86–95, 2012.

[39] U. K. Wickramasinghe, R. Carrese, and X. Li. Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1857–1864. IEEE Press, 2010.

[40] H. Yıldız and S. Suri. On Klee's measure problem for grounded boxes. In *Proceedings of the ACM Symposium on Computational Geometry (SoCG)*, pages 111–120, 2012.

[41] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Proceedings of the International Conference on Parallel Problem Solving from Nature (PPSN)*, volume 3242 of *LNCS*, pages 832–842. Springer, 2004.

[42] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[43] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.