Scalarization versus Indicator-based Selection in Multi-Objective CMA Evolution Strategies

Thomas Voß
Institut für Neuroinformatik
Ruhr-Universität Bochum
44780 Bochum, Germany
thomas.voss@neuroinformatik.rub.de

Nicola Beume
Lehrstuhl für Algorithm Engineering
Technische Universität Dortmund
44221 Dortmund, Germany
nicola.beume@tu-dortmund.de

Günter Rudolph
Lehrstuhl für Algorithm Engineering
Technische Universität Dortmund
44221 Dortmund, Germany
guenter.rudolph@tu-dortmund.de

Christian Igel
Institut für Neuroinformatik
Ruhr-Universität Bochum
44780 Bochum, Germany
christian.igel@neuroinformatik.rub.de

Abstract—**While scalarization approaches to multi-criteria optimization become infeasible in the case of many objectives, for few objectives the benefits of population-based methods compared to a set of independent single-objective optimization trials on scalarized functions are not obvious.**

The multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) is a powerful algorithm for real-valued multi-criteria optimization. This population-based approach combines mutation and strategy adaptation from the elitist CMA-ES with multi-objective selection.

We empirically compare the steady-state MO-CMA-ES with different scalarization algorithms, in which the elitist CMA-ES is used as single-objective optimizer. Although only bicriteria benchmark problems are considered, the MO-CMA-ES performs best in the overall comparison. However, if the scalarized problems have a structure that can easily be exploited by the CMA-ES and that is less apparent in the vector-valued fitness function, the CMA-ES with scalarization outperforms the population-based approach.

I. Introduction

The goal of multi-objective optimization (MOO) can be defined as efficiently finding a set of solutions that diversely and accurately approximates the set of Pareto-optimal solutions. Evolutionary algorithms (EAs) are particularly well-suited for MOO. This may have several reasons, but one of the most intriguing arguments is that the search in MOO for a set of solutions perfectly matches the concept of populations in EAs. In this study, we empirically investigate the role of the population dynamics in a recently proposed evolutionary MOO algorithm.

Many established deterministic approaches to MOO consider the scalarization of the vector-valued problem. To find a diverse approximation of the set of Pareto-optimal solutions, the vector-valued problem is reduced to a number of different scalar-valued optimization problems, which favor different trade-offs between objectives. Prominent examples of scalarization methods are the basic weighted-sum approach and the Tchebycheff method [1].

In contrast, most state-of-the-art evolutionary methods for solving MOO problems more directly rely on the notion of Pareto-dominance for finding good approximations of the Pareto-optimal set. This does not explicitly need information regarding the importance of single criteria. However, selection in general requires a ranking of the individuals in the current population including the comparison of solutions that are not related in terms of Pareto-dominance. This ranking may be viewed as some kind of scalarization, which is, however, dynamic and usually for each solution dependent on the other solutions represented in the current population. In the population-based algorithm considered here, this ranking is induced by the hypervolume in the objective space covered by the solutions in the population: Selection aims at preserving those solutions that maximize the covered hypervolume.

Scalarization approaches as outlined above are only feasible if there are only very few objectives, as the number of combinations of weighting coefficients expressing relative preferences for objectives grows exponentially with the number of objectives and so scales the number of single-objective optimization trials in simple scalarization methods. However, for two or three objectives the benefits of directly evolving a set of solutions compared to independent (parallel) single-objective optimization trials on scalarized functions are not obvious. Note that there exists scalarization methods (e.g., based on the Tchebycheff metric) that do not suffer from the often mentioned problems of weighted-sum approaches with non-convex MOO problems.

In this study, we consider the covariance matrix adaptation evolution strategy (CMA-ES [2], [3]), which is an elaborate algorithm for real-valued optimization performing well in real-world applications[1] as well as in benchmark scenarios [4], and its multi-objective variant, the MO-CMA-ES [5], [6]. The latter features a population-based approach combining the methods for mutation and strategy adaptation from the elitist CMA-ES with multi-objective selection. This makes the algorithm ideal for studying the influence of the population dynamics in the case of few objectives, because we can directly compare the MO-CMA-ES with single-objective optimizations using the elitist CMA-ES. To this end, we formulate "hybrid" algorithms combining established scalarization methods with the CMA-ES.

The remainder of this paper is organized as follows. First, two basic methods for scalarization are presented (Sec. II). In Sec. III, the single- and multi-objective elitist CMA-ESs are described and hybrid algorithms using scalarization and the CMA-ES are introduced. The experimental setup is given in Sec. IV and results of the performance assessment are presented. The paper closes with conclusions and ideas for future work.

II. Scalarization of Multi-Objective Optimization Problems

This section introduces the weighted-sum method and the method of weighted metrics for the scalarization of vector-valued optimization problems. Moreover, the ability of these methods to find Pareto-optimal solutions is briefly discussed.

Real-valued, m-objective optimization problems

$$\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m, \boldsymbol{x} \mapsto (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$$

to be minimized in each objective are considered.

A. The Weighted-Sum Method

The weighted-sum method (e.g., [7], [8]) assigns a weighting coefficient to each objective function and minimizes the weighted sum of the objectives. Thus, the vector-valued optimization problem is transformed into a scalar optimization problem of the following form:

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad \omega^{(\boldsymbol{f},\boldsymbol{w})}(\boldsymbol{x}) = \sum_{i=1}^{m} w_i f_i(\boldsymbol{x}) \ ,$$

where $\boldsymbol{w}$ is a weighting-vector with $w_i \geq 0$ for all $i = 1, \ldots, m$ and $\sum_{i=1}^{m} w_i = 1$.

B. The Method of Weighted Metrics

In the method of weighted metrics, the distance between some reference point and the feasible objective region is minimized. Most often, the utopian point $\boldsymbol{u}^*$ is used as a reference and L_p-metrics are used to measure distance. Moreover, each single objective function is associated with a weighting coefficient. Thus, the multi-objective optimization problem $\boldsymbol{f}$ is replaced with the following single-objective optimization problem, called the *weighted L_p-problem*:

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad L_p^{(\boldsymbol{f},\boldsymbol{w},\boldsymbol{u}^*)}(\boldsymbol{x}) = \left(\sum_{i=1}^{m} w_i |f_i(\boldsymbol{x}) - u_i^*|^p \right)^{1/p}$$

with $1 \leq p < \infty$, $w_i \geq 0$ for all $i = 1, \ldots, m$ and $\sum_{i=1}^{m} w_i = 1$. In the case of $p = \infty$, the metric is also called the *Tchebycheff metric* and the *weighted L_∞-* or the *Tchebycheff problem* is of the form:

$$\underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad \tau^{(\boldsymbol{f},\boldsymbol{w},\boldsymbol{u}^*)}(\boldsymbol{x}) = \max_{i=1,\ldots,m} w_i |f_i(\boldsymbol{x}) - u_i^*|$$

The utopian point $\boldsymbol{u}^* = \boldsymbol{z}^* - \varepsilon \cdot \mathbf{1}$ can be set via the *ideal point* $\boldsymbol{z}^*$ with $z_i^* = \min\{f_i(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^n\}$ and an offset $\varepsilon \cdot \mathbf{1}$, where $\varepsilon \in \mathbb{R}^+$ and $\mathbf{1}$ is the m-dimensional vector of all ones.

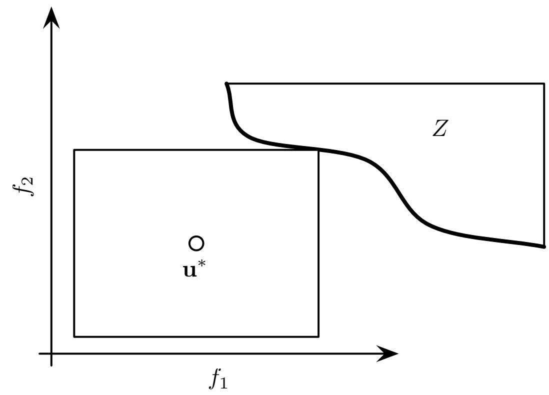

Fig. 1. Illustration of the Tchebycheff-method for the scalarization of a multi-objective optimization problem in the objective space. All valid points (Z) are reachable by a rectangle centered around an utopian point $\mathbf{u}^*$. Thus, even concave regions of the Pareto front (bold line) can be obtained.

C. Properties of Scalarization Methods

Now we review some properties of the methods introduced previously regarding their ability to find every Pareto-optimal solution of a vector-valued optimization problem.

Recall that a function $g : \mathbb{R}^n \to \mathbb{R}$ is convex if for all $\boldsymbol{x}^1, \boldsymbol{x}^2 \in \mathbb{R}^n$ and $0 \leq \beta \leq 1$ the following equation holds

$$g(\beta \boldsymbol{x}^1 + (1-\beta)\boldsymbol{x}^2) \leq \beta g(\boldsymbol{x}^1) + (1-\beta) g(\boldsymbol{x}^2) \ .$$

A set $S \subseteq \mathbb{R}^n$ is convex if $\boldsymbol{x}^1, \boldsymbol{x}^2 \in S$ implies that $\beta \boldsymbol{x}^1 + (1-\beta)\boldsymbol{x}^2 \in S$ with $0 \leq \beta \leq 1$.

A multi-objective optimization problem is convex, if all the objective functions and the feasible region are convex.

Let $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ be a multi-objective optimization problem with the feasible region $S \subset \mathbb{R}^n$. As stated in [1], if $\boldsymbol{f}$ is convex and $\boldsymbol{x}^* \in S$ is Pareto-optimal, then

[1] See the *References to CMA-ES Applications*: http://www.bionik.tu-berlin.de/user/niko/cmaapplications.pdf

there exists a weighting-vector $\boldsymbol{w} \in \mathbb{R}^m$ such that $\boldsymbol{x}^*$ is a solution of $\omega^{(\boldsymbol{f},\boldsymbol{w})}$.

For the Tchebycheff-method a stronger result holds. It is able to find every Pareto-optimal solution regardless of the vector-valued optimization problem $\boldsymbol{f}$ being convex. That is, let $\boldsymbol{x}^* \in S$ be Pareto-optimal. Then there exists a weighting vector $\boldsymbol{w} \in \mathbb{R}^m$ such that $\boldsymbol{x}^*$ is a solution of $\tau^{f,\boldsymbol{w},\boldsymbol{u}^*}$.

III. Covariance Matrix Adaptation Evolution Strategies in Multi-objective Optimization

In this section, we describe the covariance matrix adaptation evolution strategy (CMA-ES). First, we review the elitist single-objective CMA-ES, and then we outline the steady-state multi-objective CMA-ES.

The key idea of CMA-ES algorithms is to alter the mutation distribution such that the probability to reproduce steps in the search space that led to the actual population (i.e., produced offspring that were selected) is increased. This enables the algorithm to detect correlations between object variables and to become invariant under affine transformations of the search space. The algorithms implement several important concepts for adapting search distributions efficiently. The first one is known as *derandomization* meaning that the mutation distribution is adapted in a deterministic way. The second principle is *cumulation*, which refers to taking the search path of the population over the past generations into account, where the influence of previous steps decays exponentially.

A. Nomenclature

In the elitist CMA-ES, an individual $a^{(g)}$ in generation g is a 5-tuple $a^{(g)} = [\boldsymbol{x}^{(g)}, \overline{p}_{\text{succ}}^{(g)}, \sigma^{(g)}, \boldsymbol{p}_c^{(g)}, \boldsymbol{C}^{(g)}]$ comprising its candidate solution vector $\boldsymbol{x}^{(g)} \in \mathbb{R}^n$, an averaged success rate $\overline{p}_{\text{succ}}^{(g)} \in [0,1]$, the global step size $\sigma^{(g)} \in \mathbb{R}_+$, an evolution path $\boldsymbol{p}_c^{(g)} \in \mathbb{R}^n$, and the covariance matrix $\boldsymbol{C}^{(g)} \in \mathbb{R}^{n\times n}$. For the multi-objective CMA-ES, an individual $a_k^{(g)}$ denotes the k-th individual in generation g. Additionally, the following nomenclature is used:

- $f : \mathbb{R}^n \to \mathbb{R}, \boldsymbol{x} \mapsto f(\boldsymbol{x})$ is the scalar fitness function to be minimized. For the MO-CMA-ES, $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m, \boldsymbol{x} \mapsto \boldsymbol{f}(\boldsymbol{x})$ is the vector-valued fitness function.
- $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ is a multi-variate normal distribution with mean vector $\boldsymbol{m}$ and covariance matrix $\boldsymbol{C}$. The notation $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ denotes that random variable $\boldsymbol{x}$ is distributed according to $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$.
- $\boldsymbol{x}_{1:\lambda}^{(g)} \in \mathbb{R}^n$ is the best point from $\left\{\boldsymbol{x}_1^{(g)}, \dots, \boldsymbol{x}_\lambda^{(g)}\right\}$, that is, $f\left(\boldsymbol{x}_{1:\lambda}^{(g)}\right) \le f\left(\boldsymbol{x}_i^{(g)}\right)$ for all $i = 1, \dots, \lambda$. Ties are broken at random.

B. Elitist Single-Objective CMA Evolution Strategy

The (1+1)-CMA-ES combines success-based step size adaptation and elitist selection [9], [10], [11] with the covariance matrix adaptation presented in [2].

The algorithm is described within three routines. In the main routine, (1+1)-CMA-ES, the new candidate solution is sampled and the parent solution $a^{(g)}$ is updated depending on whether the new solution $a^{(g+1)}$ is better than $a^{(g)}$.

Algorithm 1: (1+1)-CMA-ES

$g = 0$, initialize $a^{(g)}$
repeat
  $a^{(g+1)} \leftarrow a^{(g)}$
  $\boldsymbol{x'}^{(g+1)} \sim \mathcal{N}\left(\boldsymbol{x}^{(g)}, {\sigma^{(g)}}^2 \boldsymbol{C}^{(g)}\right)$
  σ-`update` $\left(a^{(g+1)}, \mathbb{I}\left(f\left(\boldsymbol{x'}^{(g+1)}\right) \le f\left(\boldsymbol{x}^{(g)}\right)\right)\right)$
  if $f\left(\boldsymbol{x'}^{(g+1)}\right) \le f\left(\boldsymbol{x}^{(g)}\right)$ **then**
    $\boldsymbol{x}^{(g+1)} \leftarrow \boldsymbol{x'}^{(g+1)}$
    $\boldsymbol{C}$-`update` $\left(a^{(g+1)}, \dfrac{\boldsymbol{x}^{(g+1)} - \boldsymbol{x}^{(g)}}{\sigma^{(g)}}\right)$
  $g \leftarrow g + 1$
until stopping criterion is met

Here the indicator function $\mathbb{I}(\cdot)$ is one if its argument is true and zero otherwise. Thus, $\mathbb{I}\left(f\left(\boldsymbol{x'}^{(g+1)}\right) \le f\left(\boldsymbol{x}^{(g)}\right)\right)$ is one if the last mutation has been successful and zero otherwise. After sampling the new candidate solution, the step size is updated based on the success with a learning rate c_p $(0 < c_p \le 1)$ using a target success rate $p_{\text{succ}}^{\text{target}}$.

Procedure σ-`update`$(a = [\boldsymbol{x}, \overline{p}_{\text{succ}}, \sigma, \boldsymbol{p}_c, \boldsymbol{C}], p_{\text{succ}})$

$\overline{p}_{\text{succ}} \leftarrow (1 - c_p)\,\overline{p}_{\text{succ}} + c_p p_{\text{succ}}$

$$\sigma \leftarrow \sigma \cdot \exp\left(\frac{1}{d}\left(\overline{p}_{\text{succ}} - \frac{p_{\text{succ}}^{\text{target}}}{1 - p_{\text{succ}}^{\text{target}}}\,(1 - \overline{p}_{\text{succ}})\right)\right)$$

This update rule is rooted in the $1/5$-success-rule proposed by [9] and is an extension from the rule proposed by [12].

If the best new candidate solution was better than the parent individual (see main routine), the covariance matrix is updated as in the original CMA-ES (see [2]).

Procedure $\boldsymbol{C}$-`update`$(a = [\boldsymbol{x}, \overline{p}_{\text{succ}}, \sigma, \boldsymbol{p}_c, \boldsymbol{C}], \boldsymbol{x}_{\text{step}})$

if $\overline{p}_{\text{succ}} < p_{\text{thresh}}$ **then**
  $\boldsymbol{p}_c \leftarrow (1 - c_c)\boldsymbol{p}_c + \sqrt{c_c(2 - c_c)}\,\boldsymbol{x}_{\text{step}}$
  $\boldsymbol{C} \leftarrow (1 - c_{\text{cov}})\boldsymbol{C} + c_{\text{cov}} \cdot \boldsymbol{p}_c {\boldsymbol{p}_c}^{\mathrm{T}}$
else
  $\boldsymbol{p}_c \leftarrow (1 - c_c)\boldsymbol{p}_c$
  $\boldsymbol{C} \leftarrow (1 - c_{\text{cov}})\boldsymbol{C} + c_{\text{cov}} \cdot \left(\boldsymbol{p}_c {\boldsymbol{p}_c}^{\mathrm{T}} + c_c(2 - c_c)\boldsymbol{C}\right)$

The update of the evolution path $\boldsymbol{p}_c$ depends on the value of $\overline{p}_{\text{succ}}$. If the smoothed success rate $\overline{p}_{\text{succ}}$ is high, that is, above $p_{\text{thresh}} < 0.5$, the update of the evolution path $\boldsymbol{p}_c$ is stalled. This prevents a too fast increase of axes of $\boldsymbol{C}$ when the step size is far too small, for example, in a linear surrounding. The constants c_c and c_{cov} $(0 \le c_{\text{cov}} < c_c \le 1)$ are learning rates for the evolution path and the covariance matrix, respectively.

The factor $\sqrt{c_c(2-c_c)}$ normalizes the variance of $\boldsymbol{p}_c$ viewed as a random variable (see [2]). The evolution path $\boldsymbol{p}_c$ is used to update the covariance matrix. The new covariance matrix is a weighted mean of the old covariance matrix and the outer product of $\boldsymbol{p}_c$. In the second case (line 5), the second summand in the update of $\boldsymbol{p}_c$ is missing and the length of $\boldsymbol{p}_c$ shrinks. Although of minor relevance, the term $c_c(2-c_c)\boldsymbol{C}$ (line 6) compensates for this shrinking in $\boldsymbol{C}$.

The (external) strategy parameters are the population size, target success probability $p_{\text{succ}}^{\text{target}}$, step size damping d, success rate averaging parameter c_p, cumulation time horizon parameter c_c, and covariance matrix learning rate c_{cov}. Default values, as given in [5] and used in this paper, are:

$$
\begin{aligned}
d &= 1 + n/2 \\
p_{\text{succ}}^{\text{target}} &= \left(5 + \sqrt{1/2}\right)^{-1} \\
c_p &= p_{\text{succ}}^{\text{target}} / \left(2 + p_{\text{succ}}^{\text{target}}\right) \\
c_c &= 2/(n+2) \\
c_{\text{cov}} &= 2/(n^2+6) \\
p_{\text{thresh}} &= 0.44
\end{aligned}
$$

The elements of the initial individual $a^{(0)}$ are set to $\overline{p}_{\text{succ}} = p_{\text{succ}}^{\text{target}}$, $\boldsymbol{p}_c = \boldsymbol{0}$, and $\boldsymbol{C} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The initial candidate solution $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ and the initial $\sigma \in \mathbb{R}^+$ must be chosen problem dependent. The optimum should presumably be within the cube $\bigotimes_{i=1}^{n} [\, x_i^{(0)} - 2\,\sigma, x_i^{(0)} + 2\,\sigma\,]$.

C. Multi-Objective CMA Evolution Strategy

In the following, we briefly outline the multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) according to [6], for a detailed description and a performance evaluation on benchmark functions (in particular a comparison with real-valued NSGA-II [13]) we refer to [5].

The MO-CMA-ES relies on the non-dominated sorting selection scheme suggested by [13]. As in the SMS-EMOA [14], the hypervolume-indicator serves as second-level sorting criterion to rank individuals on the same level of non-dominance.

Let A be a population, and let a, a' be two individuals in A. Let the non-dominated solutions in A be denoted by $\text{ndom}(A) = \{a \in A | \nexists a' \in A : a' \prec a\}$ where $\prec$ denotes the Pareto-dominance relation. The elements in $\text{ndom}(A)$ are assigned a level of non-dominance of 1. The other ranks of non-dominance are defined recursively by considering the set without the solutions with lower ranks ([13]). Formally, let $\text{dom}_0(A) = A, \text{dom}_l(A) = \text{dom}_{l-1}(A) \setminus \text{ndom}_l(A)$, and $\text{ndom}_l(A) = \text{ndom}(\text{dom}_{l-1}(A))$ for $l \geq 1$. For $a \in A$ we define the level of non-dominance $\text{rank}(a, A)$ to be i iff $a \in \text{ndom}_i(A)$. Let $\Delta_{\mathcal{S}}(a, A)$ be the contributing hypervolume of a with respect to $\{a' \in A | \operatorname{rank}(a', A) = \operatorname{rank}(a, A)\}$ [14]. Moreover, let $\text{cont}(a, A)$ be the contribution rank of a with respect to its contributing hypervolume. Then the following relation is defined.

$$
\begin{aligned}
a \prec_A a' \Leftrightarrow \quad & \text{rank}(a, A) < \text{rank}(a', A) \vee \\
& (\text{rank}(a, A) = \text{rank}(a', A) \wedge \\
& \text{cont}(a, A) > \text{cont}(a', A))
\end{aligned}
$$

The standard version of the steady-state MO-CMA-ES is given in Algorithm 4.

Algorithm 4: Steady-State MO-CMA-ES

1 $g = 0$, initialize $Q^{(g)} \leftarrow \left\{ a_k^{(g)} | 1 \leq k \leq \mu \right\}$
2 **repeat**
3 $\quad i \leftarrow \mathcal{U}(1, |\,\text{ndom}\left(Q^{(g)}\right)|)$
4 $\quad a^{(g+1)} \leftarrow Q_{\prec:i}^{(g)}$
5 $\quad a'^{(g+1)} \leftarrow a^{(g+1)}$
6 $\quad \boldsymbol{x}'^{(g+1)} \sim \mathcal{N}\left(\boldsymbol{x}^{(g+1)}, \sigma^{(g+1)^2} \boldsymbol{C}^{(g+1)}\right)$
7 $\quad Q^{(g)} \leftarrow Q^{(g)} \cup \left\{ a'^{(g+1)} \right\}$
8 $\quad \boldsymbol{C}\texttt{-update}\left(a'^{(g+1)}, \dfrac{\boldsymbol{x}'^{(g+1)} - \boldsymbol{x}^{(g+1)}}{\sigma^{(g+1)}}\right)$
9 $\quad \sigma\texttt{-update}\left(a^{(g+1)}, \mathbb{I}\left(a'^{(g+1)} \prec_{Q^{(g)}} a^{(g+1)}\right)\right)$
10 $\quad \sigma\texttt{-update}\left(a'^{(g+1)}, \mathbb{I}\left(a'^{(g+1)} \prec_{Q^{(g)}} a^{(g+1)}\right)\right)$
11 $\quad Q^{(g+1)} \leftarrow Q^{(g)} \setminus Q_{\prec:\mu+1}^{(g)}$
12 $\quad g \leftarrow g + 1$
13 **until** stopping criterion is met

The parent of the single offspring is selected uniformly at random among the non-dominated solutions in the current population and an offspring is generated (line 4–7). The step size of the parent and its offspring are updated depending on whether the mutation was successful, that is, whether the offspring is better than the parent according to the relation $\prec_{Q^{(g)}}$.

Both step size and covariance matrix update are the same as in the single-objective (1+1)-CMA-ES described above.

The best μ individuals in $Q^{(g)}$ sorted by $\prec_{Q^{(g)}}$ form the next parent generation (line 11, where $Q_{\prec:i}^{(g)}$ is the i-th best offspring in $Q^{(g)}$ w.r.t. $\prec_{Q^{(g)}}$).

D. Hybrid MO-CMA-ES

In this section we introduce the canonical version of an algorithm for solving vector-valued optimization problems by using the scalarization methods discussed in Sec. II and the single-objective (1+1)-CMA-ES. Depending on the scalarization method, the algorithm is denoted by (1+1)-CMA-ES$^\tau$ or (1+1)-CMA-ES$^\omega$, where the superscript indicates whether the Tchebycheff (τ) or the weighted sum approach (ω) is used.

a) Choosing the weight vector: We restrict ourselves to bicriteria optimization problems $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^2$ that allow for a simple and efficient scheme to systematically select the weighting-vectors. Given a number of weight steps $l > 1$ a set of weighting-vectors

$$W = \{(\alpha, 1-\alpha) \mid \alpha = i/(l-1), 0 \le i \le l-1\}$$

is generated.

b) Calculating the utopian point: The Tchebycheff-method requires a utopian point $\boldsymbol{u}^*$. Unfortunately, this point is not known in advance given a multi-objective optimization problem. Thus, we rely on the (1+1)-CMA-ES to calculate an estimate of the ideal point $\boldsymbol{z}^*$ and set $\boldsymbol{u}^* = \boldsymbol{z}^* - \varepsilon \cdot \mathbf{1}$ for some $\varepsilon > 0$ and the m-dimensional vector of ones $\mathbf{1}$. Let $\boldsymbol{f}$ be a bicritera optimization problem, and let $f_1(\boldsymbol{x}^{f_1})$, $f_2(\boldsymbol{x}^{f_2})$ be the objective values of the final points found by the (1+1)-CMA-ES running on f_1 and f_2, respectively. Then we choose $z_1^* = f_1(\boldsymbol{x}^{f_1})$ and $z_2^* = f_2(\boldsymbol{x}^{f_2})$.

c) Hybrid algorithms: If Tchebycheff scalarization is used, the algorithm reads:

Algorithm 5: (1+1)-CMA-ES$^\tau$

1 initialize $\boldsymbol{x}^{(0)}$
2 $(\boldsymbol{x}^*, \boldsymbol{f}(\boldsymbol{x}^*)) \leftarrow$ (1+1)-CMA-ES on f_1
3 $\mathcal{F} \leftarrow \{\boldsymbol{f}(\boldsymbol{x}^*)\}$
4 $u_1^* = f_1(\boldsymbol{x}^*) - \varepsilon$
5 initialize $\boldsymbol{x}^{(0)}$
6 $(\boldsymbol{x}^*, \boldsymbol{f}(\boldsymbol{x}^*)) \leftarrow$ (1+1)-CMA-ES on f_2
7 $\mathcal{F} \leftarrow \mathcal{F} \cup \{\boldsymbol{f}(\boldsymbol{x}^*)\}$
8 $u_2^* = f_2(\boldsymbol{x}^*) - \varepsilon$
9 $W \leftarrow \{(\alpha, 1-\alpha) | \alpha = i/(l-1), 0 < i < l-1\}$
10 **foreach** $\boldsymbol{w} \in W$ **do**
11   initialize $\boldsymbol{x}^{(0)}$
12     $(\boldsymbol{x}^*, \boldsymbol{f}(\boldsymbol{x}^*)) \leftarrow$ (1+1)-CMA-ES on $\tau^{(\boldsymbol{f}, \boldsymbol{w}, \boldsymbol{u}^*)}$
13     $\mathcal{F} \leftarrow \mathcal{F} \cup \{\boldsymbol{f}(\boldsymbol{x}^*)\}$

In lines 1–9 a utopian point is estimated based on an estimate of the optimum for each objective.

Then for each $\boldsymbol{w} \in W$, $w_i \ge 0, i = 1, 2$ and $\sum_{i=1}^{2} w_i = 1$, the (1+1)-CMA-ES is run on $\tau^{(\boldsymbol{f}, \boldsymbol{w}, \boldsymbol{u}^*)}$ for a given number of generations (line 12), producing a solution $\boldsymbol{x}^*$ and the corresponding objective vector $\boldsymbol{f}(\boldsymbol{x}^*)$.

If the weighted-sum approach is used, $\tau^{(\boldsymbol{f}, \boldsymbol{w}, \boldsymbol{u}^*)}$ is simply replaced by $\omega^{(\boldsymbol{f}, \boldsymbol{w})}$ in the algorithm above. In the weighted sum approach, an estimate of a utopian point is not required and lines 1–9 can be replaced by $\mathcal{F} \leftarrow \emptyset$ if we set $W \leftarrow \{(\alpha, 1-\alpha) | \alpha = i/(l-1), 0 \le i \le l-1\}$.

For each single-objective optimization of a scalarized function, all parameters of the (1+1)-CMA-ES are reinitialized, in particular the global step-size σ.

However, given a weighting-vector $\boldsymbol{w}$ and the corresponding solution $\boldsymbol{x}^{\boldsymbol{w}}$, Rudolph et al. suggested in [15] to use $\boldsymbol{x}^{\boldsymbol{w}}$ as initial search point in solving the single-objective problem corresponding to $\boldsymbol{w}' = \boldsymbol{w} + (1/(l-1), 1/(l-1))$. Apart from the initial search point $\boldsymbol{x}^{(0)}$, all parameters of the (1+1)-CMA-ES are reinitialized. The modified algorithms are denoted by (1+1)-CMA-ES$^{*,\tau}$ and (1+1)-CMA-ES$^{*,\omega}$, respectively. For example, the (1+1)-CMA-ES$^{*,\omega}$ algorithm reads:

Algorithm 6: (1+1)-CMA-ES$^{*,\omega}$

$\mathcal{F} \leftarrow \emptyset$, initialize $\boldsymbol{x}^{(0)}$
$W \leftarrow \{(\alpha, 1-\alpha) | \alpha = i/(l-1), 0 \le i \le l-1\}$
foreach $\boldsymbol{w} \in W$ **do**
    $(\boldsymbol{x}^*, \boldsymbol{f}(\boldsymbol{x}^*)) \leftarrow$ (1+1)-CMA-ES on $\omega^{(\boldsymbol{f}, \boldsymbol{w})}$
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{\boldsymbol{f}(\boldsymbol{x}^*)\}$
    $\boldsymbol{x}^{(0)} \leftarrow \boldsymbol{x}^*$

IV. Experiments

The experimental setup for the performance evaluation has been chosen similar to the one in [6].

The results of both variants of the (1+1)-CMA-ES$^\tau$ and (1+1)-CMA-ES$^\omega$ were compared to results of the steady-state MO-CMA-ES using the $\mathcal{S}$-measure as second-level sorting criterion. Our evaluation is based on two performance indicators, the ϵ- and the hypervolume-indicator. We use the evaluation module of PISA[2] presented in [16] for parts of the performance assessment. We briefly outline the evaluation process below and refer to [17], [18] for a detailed description of the methods.

Before performance indicators are computed, the data are normalized. We want to compare $k = 5$ algorithms on a particular optimization problem $\boldsymbol{f}$ after g fitness evaluations and we assume that we have conducted t trials. We consider the non-dominated individuals of the union of all $k \cdot t$ populations after g evaluations. These individuals make up the reference set $\mathcal{R}$. Their objective vectors are normalized by an affine linear transformation such that for every objective the smallest and largest objective function values are mapped to 1 and 2, respectively.

The higher the value for the unary hypervolume-indicator the more volume is covered by the final set of solution and the better the performance.[3]

In contrast, a lower value of the binary ϵ-indicator means better performance. The ϵ-indicator is computed with respect to the reference set $\mathcal{R}$, and the smaller the value the "closer" is the result to $\mathcal{R}$.

A. Benchmark Functions

Several benchmark functions have been chosen from the literature. The constrained functions ZDT1, ZDT2, ZDT3, and ZDT6 [19] and their rotated variants termed IHR1, IHR3 and IHR6 [5] were selected. Additionally,

[2]PISA stands for *Platform and Programming Language Independent Interface for Search Algorithms*: http://www.tik.ee.ethz.ch/pisa/

[3]Note that in [16], [5], [6] the hypervolume is compared to a reference set and therefore a lower value indicates better performance.

the unconstrained and rotated functions ELLI_1, ELLI_2, CIGTAB_1, and CIGTAB_2 proposed in [5] were used. The benchmark function WFG1 introduced in [20] has been chosen to illustrate the ability of a scalarization method to find solutions in concave regions of the Pareto-optimal front. In Table I the definitions of the benchmark functions are summarized.

B. Parameter Setup

For each of the algorithms, 50 independent trials with 100,000 fitness function evaluations were carried out. For the hybrid algorithms a number of weight steps $l = 100$ with the single-objective CMA-ES running for 1000 generations was chosen. For the MO-CMA-ES, a population size of $\mu = 100$ and an offspring population size of $\lambda = 1$ was chosen. Accordingly, the number of generations was set to 100,000. All other parameters were set to standard values.

C. Results

Table II shows the results of the performance evaluation after 50 trials with 100,000 fitness evaluations. After 50,000 fitness evaluations the results are similar and therefore not shown here. Some differences are so small that they are not apparent in the result table, although they are statistically significant. For the function ELLI_2, where the results are very close, we additionally provide boxplots of the indicator values in Fig. 2.

D. Discussion

The MO-CMA-ES performs significantly better than the (1+1)-CMA-ES^{τ} and the (1+1)-CMA-$\text{ES}^{*,\tau}$ on all benchmark functions except for ELLI_1 and ELLI_2.

On nine test problems, (1+1)-CMA-$\text{ES}^{*,\tau}$ performs significantly better than (1+1)-CMA-ES^{τ}, while (1+1)-CMA-ES^{τ} is only significantly better than (1+1)-CMA-$\text{ES}^{*,\tau}$ on IHR6. Thus, the warm starts in the (1+1)-CMA-$\text{ES}^{*,\tau}$ improve the algorithms

The function WFG1 has a non-convex Pareto front, and thus it does not come as a surprise that the weighted-sum approach performs badly on WFG1.

The good results of the (1+1)-CMA-ES^{ω} algorithms on the functions CIGTAB_1, CIGTAB_2, ELLI_1, and ELLI_2 are striking. But these are ideal test functions for this method. First, as all objective functions are convex, the weighted-sum approach is in principle able to find all Pareto-optimal points. Second, for fixed weighting coefficients, these functions reduce to quadratic optimization problems (i.e., for these two functions $w_1 f_1(\boldsymbol{x}) + (1 - w_1) f_2(\boldsymbol{x})$ are elliptic paraboloids), and exactly these problems are ideal for the CMA-ES. That is, on CIGTAB_1, CIGTAB_2, ELLI_1, and ELLI_2, the algorithms (1+1)-CMA-ES^{ω} and (1+1)-CMA-$\text{ES}^{*,\omega}$ decompose the multi-objective optimization problem into subproblems perfectly suited for the CMA-ES, and this results in superior performance on these benchmark functions.

V. Conclusions

Scalarization approaches to multi-objective optimization (MOO) as considered in this paper suffer from the "curse of dimensionality" and are not suited for many objectives. However, for bicriteria optimization single-objective evolutionary algorithms applied to a set of scalarized fitness functions may be competitive to population-based approaches, in which the population of a single trial is considered as the solution to the MOO problem.

Here we compared the multi-objective CMA evolution strategy (MO-CMA-ES), which is based on the elitist CMA-ES, with application of the elitist CMA-ES to the optimization of scalarized fitness functions. To this end, we introduced new "hybrid" algorithms combining established scalarization methods with the CMA-ES. These algorithms have similar invariance properties as the MO-CMA-ES.

On most standard benchmark functions, the MO-CMA-ES outperformed the elitist CMA-ES combined with scalarization. But when the scalarization produced perfect quadratic fitness functions, the powerful strategy adaptation of the CMA-ES could exploit this ideal structure and the CMA-ES combined with weighted-sum scalarization performed on par with the truly population-based MO-CMA-ES. Because of the known problems of the weighted-sum approach with non-convexity, the hybrid algorithm using the Tchebycheff metric performed better compared to weighted aggregation on non-convex MOO problems.

Future work will include the evaluation of the proposed algorithms on more than two objectives with the hypothesis that the MO-CMA-ES will outperform the hybrid methods even more clearly. Further, we will compare the performance in scenarios where the number of fitness evaluations is strictly limited.

References

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

[2] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[3] N. Hansen, "The CMA evolution strategy: A comparing review," in *Towards a new evolutionary computation. Advances on estimation of distribution algorithms.*, I. I. J.A. Lozano, P. Larraãga and E. Bengoetxea, Eds. Springer-Verlag, 2005.

[4] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*. IEEE Press, 2005, pp. 1769–1776.

[5] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, no. 1, 2006.

[6] C. Igel, T. Suttorp, and N. Hansen, "Steady-state selection and efficient covariance matrix update in the multi-objective cma-es," in *Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, ser. LNCS, vol. 4403. Springer-Verlag, 2007.

TABLE I

Benchmark problems to be minimized, $\boldsymbol{y} = \boldsymbol{O}\boldsymbol{x}$, where $\boldsymbol{O} \in \mathbb{R}^{n \times n}$ is an orthogonal matrix, and $y_{\text{MAX}} = 1/\max_j(|o_{1j}|)$. For the definition of h, h_f, and h_g we refer to [19]. For the rotated benchmark functions, with $a = 1000$, $b = 100$, $\boldsymbol{y} = \boldsymbol{O}_1\boldsymbol{x}$, and $\boldsymbol{z} = \boldsymbol{O}_2\boldsymbol{x}$, where $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$ are orthogonal matrices. For the benchmark function WFG1, the input vector $\boldsymbol{x}$ is transformed several times resulting in the vector $\boldsymbol{t}^4$ with $k = 2$. We refer to [20] for the definition of the transformations.

Problem	n	Variable bounds	Objective function	Optimal solution
IHR1	10	$[-1, 1]$	$f_1(\boldsymbol{x}) = \lvert y_1 \rvert$ $f_2(\boldsymbol{x}) = g(\boldsymbol{y})\, h_f\!\left(1 - \sqrt{h(y_1)/g(\boldsymbol{y})}\right)$ $g(\boldsymbol{y}) = 1 + 9\left(\sum_{i=2}^{n} h_g(y_i)\right) / (n-1)$	$y_1 \in [0, y_{\max}]$ $y_i = 0$ $i = 2, \dots n$
IHR3	10	$[-1, 1]$	$f_1(\boldsymbol{x}) = \lvert y_1 \rvert$ $f_2(\boldsymbol{x}) = g(\boldsymbol{y})\, h_f\!\left(1 - \sqrt{h(y_1)/g(\boldsymbol{y})} - \frac{h(y_1)}{g(\boldsymbol{y})} \sin(10\pi y_1)\right)$ $g(\boldsymbol{y}) = 1 + 9\left(\sum_{i=2}^{n} h_g(y_i)\right) / (n-1)$	$y_1 \in [0, y_{\max}]$ $y_i = 0$ $i = 2, \dots n$
IHR6	10	$[-1, 1]$	$f_1(\boldsymbol{x}) = 1 - \exp(-4\lvert y_1 \rvert)) \sin^6(6\pi y_1)$ $f_2(\boldsymbol{x}) = g(\boldsymbol{y})\, h_f\!\left(1 - (f_1(\boldsymbol{x})/g(\boldsymbol{y}))^2\right)$ $g(\boldsymbol{y}) = 1 + 9\left[\left(\sum_{i=2}^{n} h_g(y_i)\right) / (n-1)\right]^{0.25}$	$y_1 \in [-y_{\max}, y_{\max}]$ $y_i = 0$ $i = 2, \dots n$
WFG1	10	$[0, 1]$	$f_1(\boldsymbol{x}) = 1 - \cos(0.5 t_1^4 \pi)$ $f_2(\boldsymbol{x}) = 1 - t_1^4 - (\cos(2 \cdot 5 \cdot \pi t_1^4 + \pi/2))/(2 \cdot 5 \cdot \pi)$	$x_i = 2i + 0.35$ $i = k+1, \dots, n$
			Unconstrained Benchmark Functions	
ELLI_1	10	$[-10, 10]$	$f_1(\boldsymbol{x}) = \frac{1}{a^2 n} \sum_{i=1}^{n} a^{2\frac{i-1}{n-1}} y_i^2$ $f_2(\boldsymbol{x}) = \frac{1}{a^2 n} \sum_{i=1}^{n} a^{2\frac{i-1}{n-1}} (y_i - 2)^2$	$y_1 = \dots = y_n$ $y_1 \in [0, 2]$
ELLI_2	10	$[-10, 10]$	$f_1(\boldsymbol{x}) = \frac{1}{a^2 n} \sum_{i=1}^{n} a^{2\frac{i-1}{n-1}} y_i^2$ $f_2(\boldsymbol{x}) = \frac{1}{a^2 n} \sum_{i=1}^{n} a^{2\frac{i-1}{n-1}} (z_i - 2)^2$	
CIGTAB_1	10	$[-10, 10]$	$f_1(\boldsymbol{x}) = \frac{1}{a^2 n}\left[y_1^2 + \sum_{i=2}^{n-1} a y_i^2 + a^2 y_n^2\right]$ $f_2(\boldsymbol{x}) = \frac{1}{a^2 n}\left[(y_1 - 2)^2 + \sum_{i=2}^{n-1} a (y_i - 2)^2 + a^2 (y_n - 2)^2\right]$	$y_1 = \dots = y_n$ $y_1 \in [0, 2]$
CIGTAB_2	10	$[-10, 10]$	$f_1(\boldsymbol{x}) = \frac{1}{a^2 n}\left[y_1^2 + \sum_{i=2}^{n-1} a y_i^2 + a^2 y_n^2\right]$ $f_2(\boldsymbol{x}) = \frac{1}{a^2 n}\left[(z_1 - 2)^2 + \sum_{i=2}^{n-1} a (z_i - 2)^2 + a^2 (z_n - 2)^2\right]$	

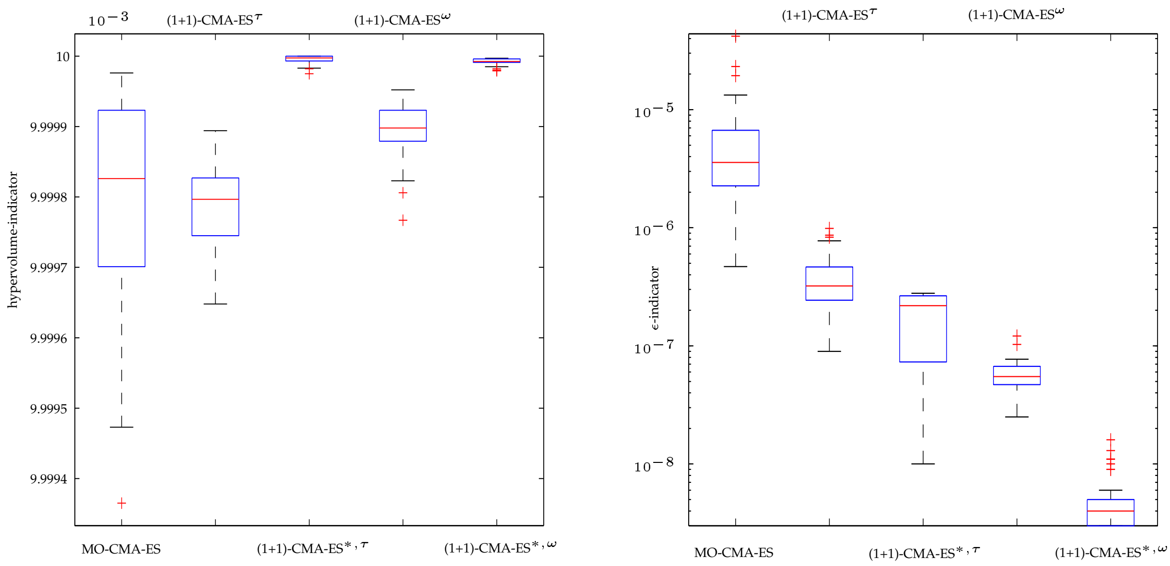

Fig. 2. Box plots of the results of the performance assessment for the fitness function ELLI_2. The left plot shows the values of the hypervolume-indicator (minus 1.2), the right plot shows the ϵ-indicator values on logarithmic scale. The boxes refer to the 25th and 75th percentile. In each plot, from left to right, results of the MO-CMA-ES, (1+1)-CMA-ES$^{\tau}$, (1+1)-CMA-ES$^{*,\tau}$, (1+1)-CMA-ES$^{\omega}$, (1+1)-CMA-ES$^{*,\omega}$ are shown.

TABLE II

RESULTS ON BENCHMARK PROBLEMS. THE UPPER AND LOWER PART OF EACH TABLE SHOWS THE MEDIAN OF 50 TRIALS AFTER 100,000 FITNESS EVALUATIONS OF THE HYPERVOLUME-INDICATOR (THE HIGHER THE BETTER) AND THE ϵ-INDICATOR (THE LOWER THE BETTER), RESPECTIVELY. THE BEST VALUE IN EACH ROW IS UNDERLINED, THE WORST IS PRINTED IN BOLD. THE SUPERSCRIPTS I, II, III, IV AND V INDICATE WHETHER AN ALGORITHM IS STATISTICALLY SIGNIFICANTLY BETTER THAN THE MO-CMA-ES, (1+1)-CMA-ES$^{\tau}$, (1+1)-CMA-ES$^{*,\tau}$, (1+1)-CMA-ES$^{\omega}$ AND (1+1)-CMA-ES$^{*,\omega}$, RESPECTIVELY (TWO-SIDED WILCOXON RANK SUM TEST, $p < 0.001$, SLANTED SUPERSCRIPTS REFER TO A SIGNIFICANCE LEVEL OF $p < 0.01$).

	MO-CMA-ES	(1+1)-CMA-ES$^{\tau}$	(1+1)-CMA-ES$^{*,\tau}$	(1+1)-CMA-ES$^{\omega}$	(1+1)-CMA-ES$^{*,\omega}$
			Hypervolume-Indicator		
CIGTAB_1	$\underline{1.12841}^{\mathrm{II,III,IV,V}}$	**1.07732**	1.12054^{II}	$1.12692^{\mathrm{II,III}}$	$1.1281^{\mathrm{II,III,IV}}$
CIGTAB_2	$\underline{1.21000}^{\mathrm{II},\mathit{III},\mathrm{IV,V}}$	$1.21000^{\mathrm{IV,V}}$	$1.21000^{\mathrm{II,IV,V}}$	**1.20999**	1.21000^{IV}
ELLI_1	**1.04238**	1.10543^{I}	1.10548^{I}	$1.10642^{\mathrm{I,II,III}}$	$\underline{1.10689}^{\mathrm{I,II,III,IV}}$
ELLI_2	1.21000	**1.21000**	$\underline{1.21000}^{\mathrm{I,II,IV,V}}$	$1.21000^{\mathrm{I,II}}$	$1.21000^{\mathrm{I,II,IV}}$
IHR1	$\underline{1.20114}$	**1.19015**	$1.19373^{\mathrm{II,IV,V}}$	1.19053	$1.19282^{\mathrm{II,IV}}$
IHR3	$1.13567^{\mathrm{II,IV}}$	**1.08831**	$1.19807^{\mathrm{I,II,IV}}$	1.11985^{II}	$\underline{1.19809}^{\mathrm{I,II,III,IV}}$
IHR6	$\underline{1.133280}^{\mathrm{II,III,IV,V}}$	0.471093^{III}	**0.270186**	$0.570920^{\mathrm{II,III}}$	$0.111020^{\mathrm{II,III,IV}}$
WFG1	$\underline{0.829085}^{\mathrm{II,III,IV,V}}$	$0.817724^{\mathrm{IV,V}}$	$0.817885^{\mathrm{IV,V}}$	0.773000^{V}	0.767089
ZDT1	$\underline{1.16089}^{\mathrm{II,III,IV,V}}$	**1.11614**	$1.16067^{\mathrm{II,IV,V}}$	1.1355^{II}	$1.14834^{\mathrm{II,IV}}$
ZDT2	$\underline{1.11247}^{\mathrm{II,III,IV,V}}$	**1.07672**	$1.08396^{\mathrm{II,IV,V}}$	1.07981^{II}	$1.08378^{\mathrm{II,IV}}$
ZDT3	$\underline{1.11447}^{\mathrm{II,III,IV,V}}$	**1.00430**	$1.10648^{\mathrm{II,IV,V}}$	1.05358^{II}	$1.08531^{\mathrm{II,IV}}$
ZDT6	$\underline{1.15017}^{\mathrm{II,III,IV,V}}$	$1.11994^{\mathrm{IV,V}}$	$1.11995^{\mathrm{II,IV,V}}$	**1.11990**	1.11993
			ϵ-Indicator		
CIGTAB_1	$\underline{2.79046 \cdot 10^{-3}}^{\mathrm{II,III,IV,V}}$	$\mathbf{9}0.92210 \cdot 10^{-3}$	$16.73330 \cdot 10^{-3\,\mathrm{II}}$	$7.75523 \cdot 10^{-3\,\mathrm{II,III}}$	$3.49994 \cdot 10^{-3\,\mathrm{II,III,IV}}$
CIGTAB_2	$1.61880 \cdot 10^{-5\,\mathrm{II,III,IV}}$	$4.04815 \cdot 10^{-5}$	$1.79580 \cdot 10^{-5\,\mathrm{II,IV}}$	$2.34480 \cdot 10^{-5\,\mathrm{II}}$	$\underline{1.60285 \cdot 10^{-5}}^{\mathrm{II,III,IV}}$
ELLI_1	$57.42470 \cdot 10^{-3}$	$5.61813 \cdot 10^{-3\,\mathrm{I}}$	$5.49933 \cdot 10^{-3\,\mathrm{I}}$	$4.07720 \cdot 10^{-3\,\mathrm{I,II,III}}$	$\underline{3.76827 \cdot 10^{-3}}^{\mathrm{I,II,III,IV}}$
ELLI_2	$35.67 \cdot 10^{-7}$	$3.22 \cdot 10^{-7\,\mathrm{I}}$	$2.19 \cdot 10^{-7\,\mathrm{I,II}}$	$0.55 \cdot 10^{-7\,\mathrm{I,II,III}}$	$\underline{0.04 \cdot 10^{-7}}^{\mathrm{I,II,III,IV}}$
IHR1	$\underline{0.457576 \cdot 10^{-2}}$	$1.696670 \cdot 10^{-2\,\mathit{IV}}$	$1.471200 \cdot 10^{-2\,\mathrm{II,IV,V}}$	$1.757260 \cdot 10^{-2}$	$1.558550 \cdot 10^{-2\,\mathrm{II,IV}}$
IHR3	$7.01850 \cdot 10^{-2\,\mathrm{II,IV}}$	$10.98640 \cdot 10^{-2}$	$1.41260 \cdot 10^{-2\,\mathrm{I,II,IV}}$	$9.28679 \cdot 10^{-2\,\mathrm{II}}$	$\underline{1.25636 \cdot 10^{-2}}^{\mathrm{I,II,III,IV}}$
IHR6	$\underline{0.563916 \cdot 10^{-1}}^{\mathrm{II,III,IV,V}}$	$6.720760 \cdot 10^{-1\,\mathrm{III}}$	$8.509980 \cdot 10^{-1}$	$5.385170 \cdot 10^{-1\,\mathrm{II,III}}$	$0.864363 \cdot 10^{-1\,\mathrm{II,III,IV}}$
WFG1	$\underline{0.496331 \cdot 10^{-2}}^{\mathrm{II,III,IV,V}}$	$3.419780 \cdot 10^{-2\,\mathrm{III,IV,V}}$	$4.519460 \cdot 10^{-2\,\mathrm{IV,V}}$	$9.892360 \cdot 10^{-2}$	$9.854170 \cdot 10^{-2\,\mathrm{IV}}$
ZDT1	$\underline{0.184177 \cdot 10^{-2}}^{\mathrm{II,III,IV,V}}$	$8.189810 \cdot 10^{-2}$	$0.746921 \cdot 10^{-2\,\mathrm{II,IV,V}}$	$4.668730 \cdot 10^{-2\,\mathrm{II}}$	$4.180090 \cdot 10^{-2\,\mathrm{II}}$
ZDT2	$\underline{0.178094 \cdot 10^{-1}}^{\mathrm{II,III,IV,V}}$	$1.170170 \cdot 10^{-1}$	$1.089780 \cdot 10^{-1\,\mathrm{II,IV,V}}$	$1.091890 \cdot 10^{-1\,\mathrm{II}}$	$1.089830 \cdot 10^{-1\,\mathrm{II,IV}}$
ZDT3	$\underline{0.13365 \cdot 10^{-2}}^{\mathrm{II,III,IV,V}}$	$15.27560 \cdot 10^{-2}$	$2.41811 \cdot 10^{-2\,\mathrm{II,IV,V}}$	$10.92340 \cdot 10^{-2\,\mathrm{II}}$	$6.113320 \cdot 10^{-2\,\mathrm{II,IV}}$
ZDT6	$\underline{1.34321 \cdot 10^{-2}}^{\mathrm{II,III,IV,V}}$	$8.13680 \cdot 10^{-2\,\mathrm{IV,V}}$	$8.13680 \cdot 10^{-2\,\mathrm{II,IV,V}}$	$8.13715 \cdot 10^{-2\,\mathrm{V}}$	$8.13797 \cdot 10^{-2}$

[7] S. Gass and T. Saaty, "The Computational Algorithm for the Parametric Objective Function," *Naval Research Logistics Quarterly*, vol. 2, no. 2, pp. 39–45, 1955.

[8] L. Zadeh, "Optimality and Non-Scalar-Valued Performance Criteria," *IEEE Transactions on Automatic Control*, vol. 8, no. 8, pp. 59–60, 1963.

[9] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, 1973.

[10] H.-P. Schwefel, *Evolution and Optimum Seeking*, ser. Sixth-Generation Computer Technology Series. John Wiley & Sons, 1995.

[11] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies: A comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

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

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

[14] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[15] G. Rudolph, B. Naujoks, and M. Preuss, "Capabilities of MOEA to Detect and Preserve Equivalent Pareto Subsets," in *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, S. Obayashi, Ed. Springer-Verlag, 2007, pp. 36–50.

[16] J. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 214, 2006, revised version.

[17] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms — a comparative case study," in *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, 1998, pp. 292–301.

[18] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca, "Performance assesment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[19] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[20] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multi-objective test problems and a scalable test problem toolkit," in *IEEE Transactions on Evolutionary Computation*, vol. 10, 2007, pp. 477–506.