Internal Report 2005–04

The Multi-objective Variable Metric Evolution Strategy Part I

by

Christian Igel, Nikolaus Hansen, and Stefan Roth

Ruhr-Universität Bochum
Institut für Neuroinformatik
44780 Bochum

IR-INI 2005–04
October 2005
ISSN 0943-2752

The Multi-objective Variable Metric Evolution Strategy Part I

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

Nikolaus Hansen nikolaus.hansen@inf.ethz.ch
Computational Science and Engineering Laboratory (CSE Lab), Swiss Federal Institute of Technology (ETH) Zurich, Switzerland

Stefan Roth stefan.roth@neuroinformatik.rub.de
Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany

The covariance matrix adaptation evolution strategy (CMA-ES) is one of the most powerful evolutionary algorithms for real-valued single-objective optimization. Here a variant of the CMA-ES for multi-objective optimization (MOO) is developed.

First a single-objective, elitist CMA-ES using plus-selection and step size control based on a success rule is introduced. This algorithm is compared to the standard CMA-ES. The elitist CMA-ES turns out to be slightly faster on unimodal functions, but is more prone to getting stuck in sub-optimal local minima.

In the new multi-objective CMA-ES (MO-CMA-ES) a population of individuals that adapt their search strategy as in the elitist CMA-ES is maintained. These are subject to multi-objective selection. The selection is based on non-dominated sorting using either the crowding-distance or the contributing hypervolume as second sorting criterion. Both the elitist single-objective CMA-ES and the MO-CMA-ES inherit important invariance properties, in particular invariance against rotation of the search space, from the original CMA-ES.

The benefits of the new MO-CMA-ES in comparison to the well-known NSGA-II and NSDE, a multi-objective differential evolution algorithm, are experimentally shown.

Keywords: Multi-objective optimization, evolution strategy, covariance matrix adaptation

1 Introduction

The covariance matrix adaptation evolution strategy (CMA-ES) is one of the most powerful evolutionary algorithms for real-valued optimization (Hansen and Ostermeier, 2001; Hansen, Müller, and Koumoutsakos, 2003; Hansen and Kern, 2004; Hansen, 2005a) with many successful applications (for an overview see Hansen, 2005b). The CMA-ES learns and employs a variable metric by means of a covariance matrix for the search distribution. The main advantages of the CMA-ES lie in its invariance properties, which are achieved by carefully designed search and selection operators, and in its efficient (self-) adaptation

of the search distribution. The CMA-ES is invariant against order-preserving transformations of the fitness function and in particular against rotation and translation of the search space—apart from the initialization. If either the strategy parameters are initialized accordingly or the time needed to adapt the strategy parameters is neglected, any affine transformation of the search space does not affect the performance of the CMA-ES. Rotation of the search space to test invariance and to generate non-separable functions was proposed by Hansen et al. (1995), and the importance of such invariance properties for evolutionary algorithms is discussed in depth by Salomon (1996) and Hansen (2000). Note that an algorithm not being invariant against a certain group of transformations means that it is biased towards a certain class of problems defined w.r.t. those transformations, for example to tasks with separable fitness functions. Such a bias is only desirable if the applications the algorithm is designed for lie in that special class. We think, for the transformations mentioned above this assumption is not attractive in general.

The interest in multi-objective optimization (MOO) is increasing rapidly. Several successful evolutionary MOO algorithms have been developed (Coello Coello, Van Veldhuizen, and Lamont, 2002; Deb, 2001), where the main focus of research has been put on the selection and archiving strategies. Multi-objective evolution strategies with (self-) adaptation of the search distribution have been proposed (Laumanns, Rudolph, and Schwefel, 2001; Büche, Müller, and Koumoutsakos, 2003; Igel, 2005), but none of them achieves the invariance properties of the single-objective CMA-ES. In this study, we therefore develop a variant of the CMA-ES for real-valued MOO.

In the CMA-ES a small population size is usually sufficient and only one set of strategy parameters is maintained. For MOO a large population is needed to evolve a diverse set of solutions, each ideally representing a (Pareto-) optimal trade-off between the objectives. The optimal strategy parameters for the members of this population may differ considerably and should therefore be adapted individually. This suggests to apply a MOO selection mechanism to a population of individuals each of which uses the strategy adaptation of the CMA-ES (for details about the covariance matrix adaptation we refer to Hansen and Ostermeier, 2001, and Hansen, 2005a). The standard single-objective CMA-ES relies on non-elitist (μ, λ)-selection, that is, the best μ of λ offspring form the next parent population and all former parents are discarded. For each set of strategy parameters to be adapted, several offspring have to be generated in each generation. If we want to maximize the number of different strategy parameter sets, given a fixed total number of offspring per iteration, the number of offspring per parent has to be as small as possible. Therefore, we first develop a single-objective, elitist CMA-ES with $(1+\lambda)$-selection, where λ can be chosen as small as one. In this elitist $(1+\lambda)$-CMA-ES the parent population consists of a single individual generating λ offspring and the best individual out of parent and offspring becomes the parent of the next generation. This $(1+\lambda)$-CMA-ES inherits all invariance properties from the original CMA-ES and is integrated into the MOO framework by considering, roughly speaking, a population of $(1+\lambda)$ evolution strategies, which are subject to multi-objective selection. Thus, the new MO-CMA-ES inherits important invariance properties from the original CMA-ES.

To summarize, the goal of this study is to augment evolutionary real-valued MOO with efficient adaptation of the search distribution and invariance against transformations of the search space. To achieve this, we develop an elitist variant of the single-objective CMA-ES. Its strategy adaptation mechanism can be combined with multi-objective selection using non-dominated sorting. To improve selection, we propose the contributing hypervolume as second sorting criterion. For better empirical evaluation, new biobjective benchmark

functions are presented. The article is organized as follows. In the next section, the new single-objective elitist (1+λ)-CMA-ES is presented and empirically compared to the standard (μ, λ)-CMA-ES. Then, in Section 3, we introduce the MO-CMA-ES using either the original selection of the non-dominated sorting genetic algorithm II (NSGA-II; Deb et al., 2002) or a new modification thereof based on the contributing hypervolume of individuals. In Section 4, the two variants of the MO-CMA-ES are empirically compared with the NSGA-II and non-dominated sorting differential evolution (NSDE; Iorio and Li, 2005). As far as we know, the latter is the only other evolutionary MOO algorithm invariant against rotation of the search space. The results substantiate our final conclusions.

2 A Single-objective Elitist CMA Evolution Strategy

In this section we combine the well know (1+λ)-selection scheme of evolution strategies (Rechenberg, 1973; Schwefel, 1995; Beyer and Schwefel, 2002) with the covariance matrix adaptation. The original update rule for the covariance matrix can be reasonably applied in the (1+λ)-selection. The cumulative step size adaptation (path length control) of the (μ/μ, λ)-CMA-ES is replaced by a success rule based step size control. The path length control cannot be easily applied, because the update of the evolution path stalls whenever no successful offspring is produced. If in this case the evolution path is long, the step size diverges.

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

$f : \mathbb{R}^n \to \mathbb{R}, \boldsymbol{x} \mapsto f(\boldsymbol{x})$ is the objective (fitness) function to be minimized.

$\lambda_{\text{succ}}^{(g+1)} = \left| \left\{ i = 1, \ldots, \lambda \,\middle|\, f(\boldsymbol{x}_i^{(g+1)}) \leq f(\boldsymbol{x}_{\text{parent}}^{(g)}) \right\} \right|$ is the number of successful new candidate solutions (successful offspring).

$\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ 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 the distribution of a random variable $\boldsymbol{x}$.

$\boldsymbol{x}_{1:\lambda}^{(g)} \in \mathbb{R}^n$ is the best point from $\boldsymbol{x}_k^{(g)}$ for $k = 1, \ldots, \lambda$, that is, $f(\boldsymbol{x}_{1:\lambda}^{(g)}) \leq f(\boldsymbol{x}_i^{(g)})$ for all $i = 1, \ldots, \lambda$.

2.1 The (1+λ)-CMA-ES

The algorithm is described within three routines. In the main routine, `(1+λ)-CMA-ES`, the λ new candidate solutions are sampled and the parent solution a_{parent} is updated depending on whether any of the new solutions is better than a_{parent}.

Algorithm 1: (1+λ)-CMA-ES

$g = 0$, initialize $a^{(g)}_{\text{parent}}$
repeat
  $a^{(g+1)}_{\text{parent}} \leftarrow a^{(g)}_{\text{parent}}$
  for $k = 1, \ldots, \lambda$ **do**
    $\boldsymbol{x}_k^{(g+1)} \sim \mathcal{N}\left(\boldsymbol{x}^{(g)}_{\text{parent}}, {\sigma^{(g)}}^2 \boldsymbol{C}^{(g)}\right)$
  `updateStepSize` $\left(a^{(g+1)}_{\text{parent}}, \dfrac{\lambda^{(g+1)}_{\text{succ}}}{\lambda}\right)$
  if $f\left(\boldsymbol{x}^{(g+1)}_{1:\lambda}\right) < f\left(\boldsymbol{x}^{(g)}_{\text{parent}}\right)$ **then**
    $\boldsymbol{x}^{(g+1)}_{\text{parent}} \leftarrow \boldsymbol{x}^{(g+1)}_{1:\lambda}$
    `updateCovariance` $\left(a^{(g+1)}_{\text{parent}}, \dfrac{\boldsymbol{x}^{(g+1)}_{\text{parent}} - \boldsymbol{x}^{(g)}_{\text{parent}}}{\sigma^{(g)}_{\text{parent}}}\right)$
  $g \leftarrow g + 1$
until stopping criterion is met

After sampling the new candidate solutions, the step size is updated based on the success rate $p_{\text{succ}} = \lambda^{(g+1)}_{\text{succ}}/\lambda$ with a learning rate c_p ($0 < c_p \leq 1$).

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

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

This update rule is rooted in the 1/5-success-rule proposed by Rechenberg (1973) and is an extension from the rule proposed by Kern et al. (2004). It implements the well-known heuristic that the step size should be increased if the success rate (i.e., the fraction of offspring better than the parent) is high, and the step size should be decreased if the success rate is low. The rule is reflected in the argument to the exponential function. For $\overline{p}_{\text{succ}} > p^{\text{target}}_{\text{succ}}$ the argument is greater than zero and the step size increases; for $\overline{p}_{\text{succ}} < p^{\text{target}}_{\text{succ}}$ the argument is smaller than zero and the step size decreases; for $\overline{p}_{\text{succ}} = p^{\text{target}}_{\text{succ}}$ the argument becomes zero and no change of σ takes place.

The argument to the exponential function is always smaller than $1/d$. It is also larger than $-1/d$ if $p^{\text{target}}_{\text{succ}} < 0.5$ (a necessary assumption). Therefore, the damping parameter d controls the rate of the step size adaptation. Using $\overline{p}_{\text{succ}}$ instead of the input argument $p_{\text{succ}} = \lambda^{(g+1)}_{\text{succ}}/\lambda$ primarily smoothes the single step size changes and has only a minor influence on the maximal possible step size changing rate.

If the best new candidate solution was successful (see main routine), the covariance matrix is updated as in the (1,λ)-CMA-ES (see Hansen and Ostermeier, 2001).

Table 1. Default parameters for the (1+λ)-CMA Evolution Strategy.

Selection:
$\lambda = 1$
Step size control:
$d = 1 + \frac{n}{2\,\lambda}$, $p_{\text{succ}}^{\text{target}} = \frac{1}{5 + \sqrt{\lambda}/2}$, $c_p = \frac{p_{\text{succ}}^{\text{target}}\,\lambda}{2 + p_{\text{succ}}^{\text{target}}\,\lambda}$
Covariance matrix adaptation:
$c_c = \frac{2}{n+2}$, $c_{\text{cov}} = \frac{2}{n^2+6}$, $p_{\text{thresh}} = 0.44$

Procedure `updateCovariance`$(a = [\boldsymbol{x}, \overline{p}_{\text{succ}}, \sigma, \boldsymbol{p}_c, \boldsymbol{C}], \boldsymbol{x}_{\text{step}} \in \mathbb{R}^n)$

1 **if** $\overline{p}_{\text{succ}} < p_{\text{thresh}}$ **then**
2 $\quad \boldsymbol{p}_c \leftarrow (1 - c_c)\boldsymbol{p}_c + \sqrt{c_c(2 - c_c)}\ \boldsymbol{x}_{\text{step}}$
3 $\quad \boldsymbol{C} \leftarrow (1 - c_{\text{cov}})\boldsymbol{C} + c_{\text{cov}} \cdot \boldsymbol{p}_c \boldsymbol{p}_c^{\mathrm{T}}$
4 **else**
5 $\quad \boldsymbol{p}_c \leftarrow (1 - c_c)\boldsymbol{p}_c$
6 $\quad \boldsymbol{C} \leftarrow (1 - c_{\text{cov}})\boldsymbol{C} + c_{\text{cov}} \cdot \left(\boldsymbol{p}_c \boldsymbol{p}_c^{\mathrm{T}} + c_c(2 - c_c)\boldsymbol{C}\right)$

The update of the evolution path $\boldsymbol{p}_c$ depends on the value of $\overline{p}_{\text{succ}}$ (here the smoothing of $\lambda_{\text{succ}}/\lambda$ is of considerable relevance). 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. If the smoothed success rate $\overline{p}_{\text{succ}}$ is low, the update of $\boldsymbol{p}_c$ is accomplished obeying an exponential smoothing. The constants c_c and c_{cov} ($0 \leq c_{\text{cov}} < c_c \leq 1$) 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 Hansen and Ostermeier, 2001). 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 $\boldsymbol{C}$.

Strategy Parameters The (external) strategy parameters are offspring number λ, 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 are given in Table 1. Most default values are derived from the precursor algorithms and validated by sketchy simulations on simple test functions: the target success rate is close to the well-known 1/5 and depends on λ, because the optimal success rate in the (1+λ)-ES certainly decreases with increasing λ. The parameters for the covariance matrix adaptation are similar to those for the (1,λ)-CMA-ES.

Initialization The elements of the initial individual, $a_{\text{parent}}^{(0)}$ are set to $\overline{p}_{\text{succ}} = p_{\text{succ}}^{\text{target}}$, $\boldsymbol{p}_c = \boldsymbol{0}$, and $\boldsymbol{C} = \mathbf{I}$, where $p_{\text{succ}}^{\text{target}}$ is given in Table 1. The initial candidate solution $\boldsymbol{x} \in \mathbb{R}^n$ and

Table 2. Single-objective test functions to be minimized, where $\boldsymbol{y} = \boldsymbol{O}\boldsymbol{x}$ and $\boldsymbol{O}$ is an orthogonal matrix, implementing an angle-preserving linear transformation

Name	Function	Initial region
Linear	$f_{\text{linear}}(\boldsymbol{x}) = y_1$	$\boldsymbol{O}^{-1}[6000, 6006]^n$
Sphere	$f_{\text{sphere}}(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$	$\boldsymbol{O}^{-1}[-1, 5]^n$
Ellipsoid	$f_{\text{elli}}(\boldsymbol{x}) = \sum_{i=1}^{n} \left(1000^{\frac{i-1}{n-1}} y_i\right)^2$	$\boldsymbol{O}^{-1}[-1, 5]^n$
Rastrigin	$f_{\text{rastrigin}}(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} \left(y_i^2 - 10\cos(2\pi y_i)\right)$	$\boldsymbol{O}^{-1}[-1, 5]^n$

the initial $\sigma \in \mathbb{R}_+$ must be chosen problem dependent. The optimum should presumably be within the cube $[\boldsymbol{x} - \sigma(1, \ldots, 1)^{\mathrm{T}}, \boldsymbol{x} + \sigma(1, \ldots, 1)^{\mathrm{T}}]$.

2.2 Simulation of the (1+λ)-CMA-ES

Test functions To validate essential properties of the search algorithm we use the single-objective test problems summarized in Table 2. The linear function f_{linear} tests the ability and the speed to increase the step size σ. On f_{sphere} basic convergence properties and the speed of step size decrease are tested. On f_{elli} the performance of the CMA procedure, that is, the ability to adapt the distribution shape to the function topography is examined. On $f_{\text{rastrigin}}$, the ability to circumvent local optima is examined. Apart from f_{linear}, the optimum function value is zero for all functions. The experimental results are independent of angle-preserving transformations, like translation and rotation of the search space, that is, they are in particular independent of the chosen orthogonal transformation matrix $\boldsymbol{O}$.

Methods We conducted 51 runs for each function and each dimension. The initial candidate solution $\boldsymbol{x}$ is chosen uniformly randomly in the initial region from Table 2, and the initial $\sigma = 3$ is half of the width of the initial interval. Excepting f_{linear}, the simulation is stopped when function value differences do not exceed 10^{-12} or when the function value becomes smaller than the target function value 10^{-9}. To conduct statistical testing the runs were ranked. Runs that reached the target function value were regarded as better and ranked according to their number of function evaluations. The remaining runs were ranked according to their final function value. To evaluate statistical significance the non-parametric Mann-Whitney U-test (Wilcoxon rank sum test) was conducted. If not stated otherwise discussed differences are significant with $p < 10^{-3}$.

Results and Discussion The (1+λ)-CMA-ES is compared to the (μ/μ_{W},λ)-CMA-ES, the standard CMA-ES with weighted global intermediate (μ/μ_{W}) recombination as described by Hansen and Kern (2004). The former is elitist and has a success rule based step size adaptation. The latter is non-elitist, uses the cumulative step size adaptation (path length control), and conducts weighted recombination of all $\mu = \lfloor \lambda/2 \rfloor$ parents.

On f_{linear} the step size increases linearly on the log-scale in all strategy variants, a minimal necessary demand on step size control (Hansen, 2006). The mean number of function evaluations to increase the step size by one order of magnitude is shown in Table 3 for two plus- and two comma-strategies. The success rule in the plus-strategy is up to five times faster than the path length control in the comma-strategies, but this difference should be usually irrelevant.

Table 3. Mean number of function evaluations needed to increase the step size by a factor of ten on f_{linear}

n	λ	(1+1)	(1+λ)	(1,λ)	(μ/μ_{W},λ)
5	8	25	60	98	72
20	12	71	128	383	222

Runs on f_{sphere}, f_{elli}, and $f_{\text{rastrigin}}$ are shown in Figure 1. First, we discuss the comparison between (1+λ)- and (1,λ)-CMA-ES for the same λ ($\square$ and $\diamond$ in the figure). On f_{sphere} and f_{elli} the two strategies perform quite similar. The slight differences on f_{sphere} are primarily the result of different step size change rates. The reason for the slight differences on f_{elli} is presumably the smaller target step size of the success based adaptation rule. A smaller step size can lead to a more pronounced evolution path that assists a faster adaptation of the covariance matrix. Both strategies perform identical on $f_{\text{rastrigin}}$ in 5D, but in 20D non-elitist (1,λ) finds significantly better solutions. The reasons are probably the advantage of the comma-selection scheme in escaping local optima and the larger adapted step sizes.

More pronounced differences can be observed between the default variants (1+1) and (μ/μ_{W},λ). On f_{sphere} and f_{elli} elitist (1+1) is roughly 1.5 times faster than (μ/μ_{W},λ). On $f_{\text{rastrigin}}$ the standard (μ/μ_{W},λ) finds the considerably (and significantly) better solutions. Here, the performance of the plus-strategy can be considerably improved if the step size change rate is slowed down by increasing the damping d, but the performance of the (μ/μ_{W},λ) cannot be achieved.

The empirical results give evidence that the plus-selection is effectively combined with the covariance matrix adaptation. On the one hand, the plus-selection together with the success rule based adaptation for the step size makes the evolution strategy faster by a factor of about 1.5 on unimodal functions. On the other hand, the comma-strategy is less susceptible to get trapped into sub-optimal local minima for two reasons. First, even a particularly well evaluated individual is abandoned in the next generation; second, the path length control adapts larger step lengths, in particular within the recombinant strategy variant (the default one).

3 Covariance Matrix Adaptation for Multi-objective Optimization

Based on the (1+λ)-CMA-ES we propose a multi-objective evolution strategy. After a brief introduction to evolutionary multi-objective optimization, we present the considered selection mechanisms, which are based on non-dominated sorting. We propose an alternative ranking of individuals that have the same level of non-dominance. The ranking relies on the contributing hypervolume and can be computed efficiently for two objectives. Then the (1+λ)-MO-CMA-ES is described.

3.1 Multi-objective Optimization

Consider an optimization problem with M objectives $f_1, \ldots, f_M : X \to \mathbb{R}$ to be minimized. The vector $f(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x}))$ is the objective vector of $\boldsymbol{x} \in X$ living in the objective space $\mathbb{R}^M$. The elements of X can be partially ordered using the concept of

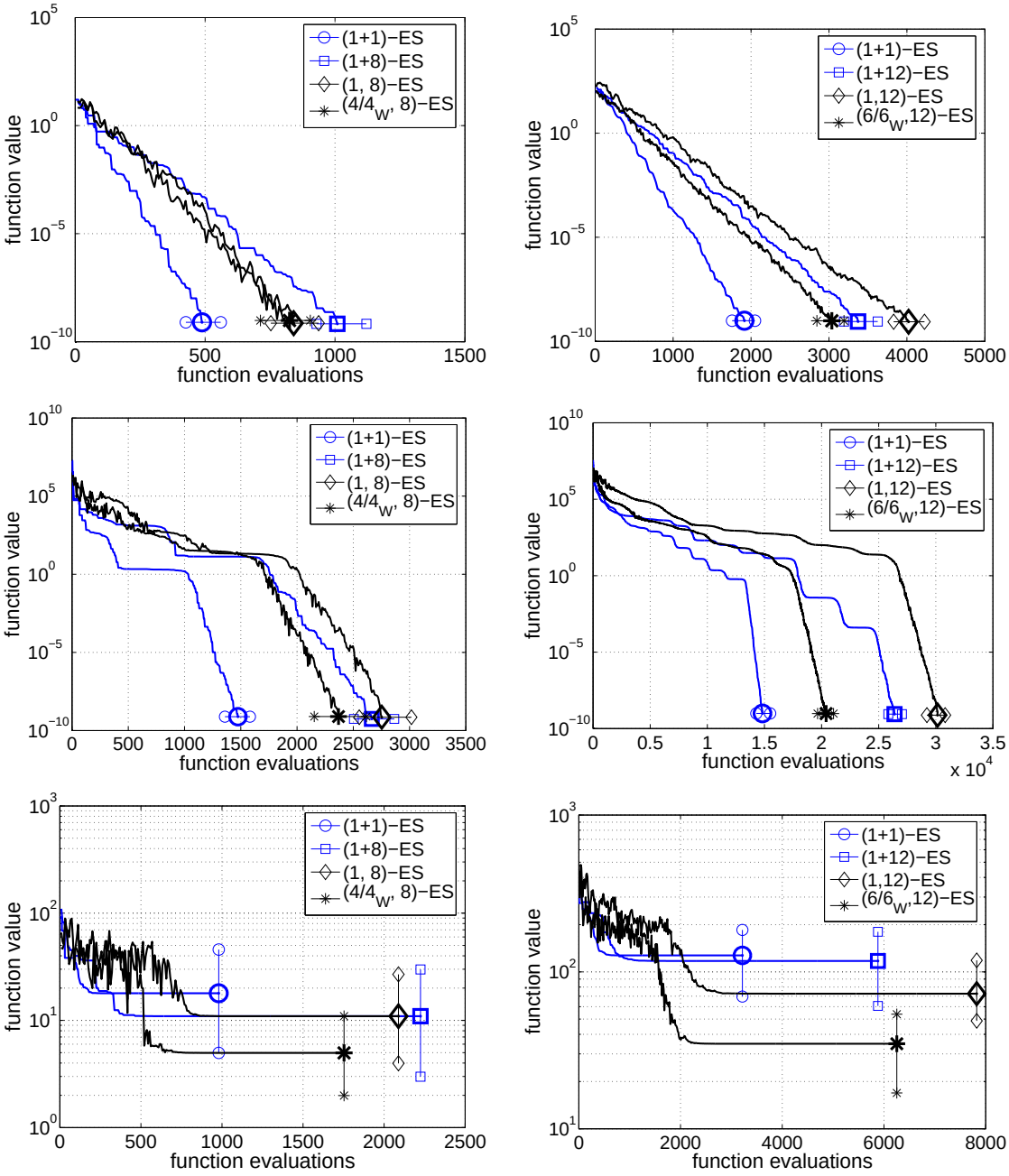

Figure 1. Simulations on Sphere (above), Ellipsoid (middle), and Rastrigin function (below), in 5D ($n = 5$, left) and 20D ($n = 20$, right). Shown is the median out of 51 runs for the (1+1)-, (1+λ)-, (1,λ)-, and (μ/μ_{W},λ)-CMA-ES. The error bars denote final values for the 3rd and the 49th run (5%- and 95%-percentile).

Pareto dominance. A solution $\boldsymbol{x} \in X$ dominates a solution $\boldsymbol{x}'$ and we write $\boldsymbol{x} \prec \boldsymbol{x}'$ iff $\forall m \in \{1, \ldots, M\} : f_m(\boldsymbol{x}) \leq f_m(\boldsymbol{x}')$ and $\exists m \in \{1, \ldots, M\} : f_m(\boldsymbol{x}) < f_m(\boldsymbol{x}')$. The elements of the (Pareto) set $\{\boldsymbol{x} \,|\, \nexists \boldsymbol{x}' \in X : \boldsymbol{x}' \prec \boldsymbol{x}\}$ are called Pareto optimal. The corresponding Pareto front is given by $\{f(\boldsymbol{x}) \,|\, \nexists \boldsymbol{x}' \in X : \boldsymbol{x}' \prec \boldsymbol{x}\} \subset \mathbb{R}^M$.

Without any further information no Pareto-optimal solution can be said to be superior

to another. The goal of multi-objective optimization (MOO) is to find a diverse set of Pareto-optimal solutions, which provide insights into the trade-offs between the objectives. When approaching an MOO problem by linearly aggregating all objectives into a scalar function, each weighting of the objectives yields only a subset of Pareto-optimal solutions (usually only a single solution). That is, various trials with different aggregations become necessary. Even worse, no linear aggregate exists such that concave parts of the Pareto front become optimal. Therefore, various trials cannot help in case of partially concave Pareto fronts (cf. Das and Dennis, 1997). Consequently, evolutionary multi-objective algorithms have become the method of choice for MOO (Coello Coello et al., 2002; Deb, 2001).

In the following, we consider evolutionary real-valued MOO, where each individual $a_i^{(g)}$ at generation g represents a real-valued candidate solution $\boldsymbol{x}_i^{(g)} \in X \subseteq \mathbb{R}^n$ of an n-dimensional problem with M objectives. For simplicity, we do not distinguish between $f_m(a_i^{(g)})$ and $f_m(\boldsymbol{x}_i^{(g)})$.

3.2 Multi-objective Selection

Our multi-objective algorithm is based on the non-dominated sorting approach used in NSGA-II (Deb, 2001; Deb et al., 2002). The individuals are sorted according to their level of non-dominance. To rank individuals on the same level an additional sorting criterion is needed. We consider two criteria, the crowding-distance and the contributing hypervolume.

3.2.1 Non-dominated Sorting First of all, the elements in a population A of candidate solutions are ranked according to their level of non-dominance. Let the non-dominated solutions in A be denoted by $\mathrm{ndom}(A) = \{a \in A \,|\, \nexists a' \in A : a' \prec a\}$. The Pareto front of A is then given by $\{(f_1(a), \ldots, f_M(a)) \,|\, a \in \mathrm{ndom}(A)\}$. The elements in $\mathrm{ndom}(A)$ get rank 1. The other ranks are defined recursively by considering the set without the solutions with lower ranks. Formally, let $\mathrm{dom}_l(A) = \mathrm{dom}_{l-1}(A) \setminus \mathrm{ndom}_l(A)$ and $\mathrm{ndom}_l(A) = \mathrm{ndom}(\mathrm{dom}_{l-1}(A))$ for $l \in \{1, \ldots\}$ with $\mathrm{dom}_0 = A$. For $a \in A$ we define the level of non-dominance $\mathrm{r}(a, A)$ to be i iff $a \in \mathrm{ndom}_i(A)$. The time complexity of non-dominated sorting of N elements is $\mathcal{O}(MN^2)$ (Deb et al., 2002).

A second sorting criterion is needed to rank the solutions having the same level of non-dominance. This criterion is very important, as usually in real-valued optimization after some generations there are more non-dominated solutions in the population than solutions to be selected. We consider two alternative additional sorting criteria, the crowding-distance (Deb et al., 2002) and the contributing hypervolume (Emmerich, Beume, and Naujoks, 2005).

3.2.2 Crowding-distance In the NSGA-II, non-dominated solutions A' with the same level of non-dominance are ranked according to how much they contribute to the spread (or diversity) of objective function values in A'. This can be measured by the crowding-distance. For M objectives, the crowding-distance of $a \in A'$ is given by

$$c(a, A') = \sum_{m=1}^{M} c_m(a, A')/(f_m^{\max} - f_m^{\min}) \ ,$$

where $f_m^{\max}$ and $f_m^{\min}$ are (estimates of) the minimum and maximum value of the mth objective and

$$c_m(a, A') := \begin{cases} \infty\,, \text{ if } f_m(a) = \min\{f_m(a') \,|\, a' \in A'\} \text{ or } f_m(a) = \max\{f_m(a') \,|\, a' \in A'\} \\ \min\{f_m(a'') - f_m(a') \,|\, a', a'' \in A' : f_m(a') < f_m(a) < f_m(a'')\}\,, \text{ otherwise.} \end{cases}$$

Based on the level of non-dominance and the crowding-distance we define the relation

$$a \prec_{c,A'} a' \Leftrightarrow \mathrm{r}(a, A') < \mathrm{r}(a', A') \text{ or } \left[(\mathrm{r}(a, A') = \mathrm{r}(a', A')) \wedge (c(a, \mathrm{ndom}_{\mathrm{r}(a',A')}(A')) > c(a', \mathrm{ndom}_{\mathrm{r}(a',A')}(A')))\right]\,,$$

for $a, a' \in A'$. That is, a is better than a' when compared using $\prec_{c,A'}$ if either a has a better (lower) level of non-dominance or a and a' are on the same level but a is in a "lesser crowded region of the objective space" and therefore induces more diversity.

The crowding-distance of N non-dominated solutions can be computed efficiently in $\mathcal{O}(MN \log N)$ (Deb et al., 2002). However, the crowding-distance is related to the spread of solutions, which may be a desirable quantity and foster evolvability, but it is not directly related to progress in terms of selecting better solutions as we will discuss in section 4.1.

3.2.3 Contributing Hypervolume The hypervolume measure or $\mathcal{S}$-metric was introduced by Zitzler and Thiele (1998) in the domain of evolutionary MOO. It can be defined as the Lebesgue measure Λ of the union of hypercubes in the objective space (Coello Coello, Van Veldhuizen, and Lamont, 2002):

$$\mathcal{S}_{a_{\mathrm{ref}}}(A') = \Lambda\left(\bigcup_{a \in \mathrm{ndom}(A')} \{(f_1(a'), \ldots, f_M(a')) \,|\, a \prec a' \prec a_{\mathrm{ref}}\}\right)\,,$$

where a_{ref} is an appropriately chosen reference point. The contributing hypervolume of a point $a \in \mathrm{ndom}(A')$ is given by

$$\Delta_{\mathcal{S}}(a, A') := \mathcal{S}_{a_{\mathrm{ref}}}(A') - \mathcal{S}_{a_{\mathrm{ref}}}(A' \setminus \{a\})\,.$$

The contributing hypervolume was used for selection in the steady-state evolutionary algorithm proposed by Emmerich, Beume, and Naujoks (2005). We adopt it, to our knowledge for the first time, for the ranking of a whole population.

Now the rank $s(a, A')$ of an individual a can be defined recursively based on its contribution to the hypervolume. The individual contributing least to the hypervolume of A' is assigned the worst rank (ties are broken at random). The individual contributing least to the hypervolume of A' without the individual with the worst rank is assigned the second worst rank and so on. Let a lower rank be worse. Formally, for $a \in \mathrm{ndom}(A')$ we have $s(a, A') = 1$ if $a = \mathrm{argmin}_{a' \in A'}\{\Delta_{\mathcal{S}}(a', A')\}$ and $s(a, A') = n$ if $a = \mathrm{argmin}_{a' \in A'}\{\Delta_{\mathcal{S}}(a', A' \setminus \{a'' \,|\, s(a'', A') < n\})\}$. The reference point a_{ref} is (implicitly) chosen in a way such that individuals a with $f_m(a) = \min\{f_m(a') \,|\, a' \in A'\}$ for any $m \in \{1, \ldots, M\}$ get the best ranks. That is, the individuals at the "edges" of the Pareto front of A' are preferably selected.

For two objectives, this ranking can be calculated efficiently in superlinear time in the number of individuals using appropriate data structures and the equation for $\Delta_{\mathcal{S}}(a, A')$ given by Emmerich et al. (2005). Unfortunately, the scaling behavior in the number of objectives is likely to be bad (While, 2005).

Lemma 1. *For two objectives, the ranks $s(a, A')$ of all individuals $a \in A'$, $|A'| = N$, can be computed in $\mathcal{O}(N \log N)$ time.*

Proof. In the following, we describe an algorithm that computes the ranking in superlinear time by storing the relevant information in appropriate data structures. We consider sorted indexed lists F and S containing individuals sorted by first fitness value and by contributing hypervolume, respectively. Consider the list S containing an individual a. Then $S[l]$ returns the lth element of S, $\text{index}(S, a)$ gives the number of a in the list (i.e., $S[\text{index}(S, a)] = a$), and $\text{insert}(S, a)$ adds and $\text{delete}(S, a)$ removes a from S. We presume an appropriate data structure (say, an AVL-tree, e.g., Knuth, 1973) that allows these look-up, insertion, and deletion operations in $\mathcal{O}(\log N)$ time, where N is the number of elements in the list.

First, S and F are filled with the elements in A'. This can be done in $\mathcal{O}(N \log N)$, because the contributing hypervolume of an individual a to the hypervolume of a set B can be computed by

$$\Delta_{\mathcal{S}}(a, B) = \begin{cases} (f_1(a_{\text{ref}}) - f_1(a)) \cdot (f_2(F[\text{index}(F, a) - 1] - f_2(a)) & \text{if } \text{index}(B, a) = |B| \\ (f_1(F[\text{index}(F, a) + 1]) - f_1(a)) \cdot (f_2(a_{\text{ref}}) - f_2(a)) & \text{if } \text{index}(B, a) = 1 \\ (f_1(F[\text{index}(F, a) + 1]) - f_1(a)) \cdot (f_2(F[\text{index}(F, a) - 1] - f_1(a)) & \text{otherwise} \end{cases}$$

in case of two objectives after the elements of B have been inserted into the list F sorted by their first fitness value (Emmerich, Beume, and Naujoks, 2005).

The elements $S[|A' - 1|]$ and $S[|A'|]$, those with the extreme f_1 values, get the ranks $|A' - 1|$ and $|A'|$. Then, $l \leftarrow 1$ and the following procedure is repeated $|A'| - 2$ times.

We determine $a \leftarrow S[1]$, the element contributing least to the hypervolume, and its neighbors in F by looking up $i \leftarrow \text{index}(F, a)$, and $a_{-1} \leftarrow F[i-1]$ and $a_{+1} \leftarrow F[i+1]$. Note that a_{-1} and a_{+1} exist, because the elements with the extreme f_1 values have maximum contributing hypervolume. The individual a is assigned the rank l, $s(a, A') \leftarrow l$, and is deleted from both lists, $\text{delete}(S, a)$ and $\text{delete}(F, a)$. We set $l \leftarrow l + 1$. The elements a_{+1} and a_{-1} are deleted from S, $\text{delete}(S, a_{-1})$ and $\text{delete}(S, a_{+1})$. The contributing hypervolumes are recomputed for a_{+1} and a_{-1} using the equation given above and the elements are reinserted into S according to the new contributing hypervolumes, $\text{insert}(S, a_{-1})$ and $\text{insert}(S, a_{+1})$.

All operations in this loop can be done in constant or logarithmic time, which proves the lemma. □

Based on this ranking and the level of non-dominance we define the relation

$$a \prec_{s,A} a' \Leftrightarrow \text{r}(a, A) < \text{r}(a', A) \text{ or } \left[(\text{r}(a, A) = \text{r}(a', A)) \wedge (s(a, \text{ndom}_{\text{r}(a',A)}(A)) > s(a', \text{ndom}_{\text{r}(a',A)}(A)))\right] ,$$

for $a, a' \in A$. That is, a is better than a' when compared using $\prec_{s,A}$ if either a has a better level of non-dominance or a and a' are on the same level but a contributes more to the hypervolume when considering the points at that level of non-dominance.

3.3 MO-CMA-ES

Now we have all ingredients for a multi-objective CMA-ES. In the $\lambda_{\text{MO}}\times(1+\lambda)$-MO-CMA-ES, we maintain a population of λ_{MO} elitist $(1+\lambda)$-CMA-ES. The kth individual in generation g is denoted by $a_k^{(g)} = [\boldsymbol{x}_k^{(g)}, \overline{p}_{\text{succ},k}^{(g)}, \sigma_k^{(g)}, \boldsymbol{p}_{c,k}^{(g)}, \boldsymbol{C}_k^{(g)}]$. For simplicity, we consider only the standard case $\lambda = 1$. The extension to $\lambda > 1$ is straightforward.

In every generation g each of the λ_{MO} parents generates $\lambda = 1$ offspring. Parents and offspring form the set $Q^{(g)}$. The step sizes of a parent and its offspring are updated depending on whether the mutations were successful, that is, whether the offspring is better than the parent according to the relation $\prec_{Q^{(g)}}$. The covariance matrix of the offspring is updated taking into account the mutation that has led to its genotype. Both step size and covariance matrix update are the same as in the single-objective $(1+\lambda)$-CMA-ES. The best λ_{MO} individuals in $Q^{(g)}$ sorted by $\prec_{Q^{(g)}}$ form the next parent generation.

Putting all together, the $\lambda_{\text{MO}}\times(1+1)$-MO-CMA reads:

Algorithm 4: $\lambda_{\text{MO}}\times(1+1)$-MO-CMA

$g = 0$, initialize $a_k^{(g)}$ for $k = 1, \ldots, \lambda_{\text{MO}}$
repeat
  for $k = 1, \ldots, \lambda_{\text{MO}}$ **do**
    $a'^{(g+1)}_k \leftarrow a_k^{(g)}$
    $\boldsymbol{x}'^{(g+1)}_k \sim \mathcal{N}\left(\boldsymbol{x}_k^{(g)}, {\sigma_k^{(g)}}^2 \boldsymbol{C}_k^{(g)}\right)$
  $Q^{(g)} = \left\{a'^{(g+1)}_k, a_k^{(g)} \mid 1 \le k \le \lambda_{\text{MO}}\right\}$
  for $k = 1, \ldots, \lambda_{\text{MO}}$ **do**
    `updateStepSize` $\left(a_k^{(g)}, \lambda^{(g+1)}_{\text{succ},Q^{(g)},k}\right)$
    `updateStepSize` $\left(a'^{(g+1)}_k, \lambda^{(g+1)}_{\text{succ},Q^{(g)},k}\right)$
    `updateCovariance` $\left(a'^{(g+1)}_k, \dfrac{\boldsymbol{x}'^{(g+1)}_k - \boldsymbol{x}_k^{(g)}}{\sigma_k^{(g)}}\right)$
  for $i = 1, \ldots, \lambda_{\text{MO}}$ **do**
    $a_i^{(g+1)} \leftarrow Q^{(g)}_{\prec:i}$
  $g \leftarrow g + 1$
until stopping criterion is met

Here

$\lambda^{(g+1)}_{\text{succ},Q^{(g)},k} = \begin{cases} 1 & \text{, if } a'^{(g+1)}_k \prec_{Q^{(g)}} a_k^{(g)} \\ 0 & \text{, otherwise} \end{cases}$ is the number of successful offspring from parent $a_k^{(g)}$ for $\lambda = 1$ and

$Q^{(g)}_{\prec:i}$ is the ith best offspring in $Q^{(g)}$ w.r.t. $\prec_{Q^{(g)}}$.

We consider two variants of the MO-CMA-ES, the c-MO-CMA and the s-MO-CMA, which use the crowding-distance and the contributing hypervolume as second level sorting criterion, respectively. That is, $\prec_{Q^{(g)}} := \prec_{c,Q^{(g)}}$ in the c-MO-CMA and $\prec_{Q^{(g)}} := \prec_{s,Q^{(g)}}$ in the s-MO-CMA, see Section 3.2.

Handling Box Constraints Consider an optimization problem with M objectives $f_1, \ldots, f_M : X \to \mathbb{R}$ with $X = [x_1^l, x_1^u] \times \cdots \times [x_n^l, x_n^u] \subset \mathbb{R}^n$. For $\boldsymbol{x} \in \mathbb{R}^n$ let

$$\text{feasible}(\boldsymbol{x}) = (\min(\max(x_1, x_1^l), x_1^u), \ldots, \min(\max(x_n, x_n^l), x_n^u))^{\mathrm{T}} \quad .$$

We define the penalized fitness

$$f_m^{\text{penalty}}(\boldsymbol{x}) = f_m(\text{feasible}(\boldsymbol{x})) + \alpha \|\boldsymbol{x} - \text{feasible}(\boldsymbol{x}))\|_2^2$$

where $\alpha > 0$ is a penalty parameter.

When in this study the MO-CMA-ES is applied to problems with box constraints the penalized fitness functions f_m^{penalty} with $\alpha = 10^{-6}$ are used in the evolutionary process.

4 Empirical Evaluation of the MO-CMA-ES

In this section, we demonstrate how the MO-CMA behaves on test functions. First, we discuss performance assessment of stochastic multi-objective algorithms in general and introduce the performance indicators. Then we empirically compare the c-MO-CMA, the s-MO-CMA, the NSGA-II, and the differential evolution method NSDE on common and new benchmark problems.

4.1 Evaluating the Performance of MOO Algorithms

The performance assessment of stochastic multi-objective algorithms is more difficult than evaluating single-objective algorithms: In empirical investigations, sets of sets, the non-dominated solutions evolved in multiple trials of different algorithms, have to be compared. Many ways of measuring the performance of MOO algorithms have been proposed. In this study, we follow recommendations by Fonseca et al. (2005, see also Knowles, Thiele, and Zitzler, 2005). We concisely define the performance measures used, for a detailed description of the methods we refer to the literature (Knowles and Corne, 2002; Zitzler et al., 2003; Knowles et al., 2005).

Given two sets of solutions $A, B \subseteq X$ there is a common sense definition of one set being better than the other. Set A is better than B and we write $A \triangleright B$ if for every element $a \in B$ there exists an element $a' \in A$ that is not worse than a in each objective, $\forall m \in \{1, \ldots, M\}, \forall a \in B, \exists a' \in A : f_m(a') \leq f(a)$, and $\text{ndom}(A) \neq \text{ndom}(B)$. Otherwise we have $A \not\triangleright B$. Regularly for two sets, A and B, neither $A \triangleright B$ nor $B \triangleright A$ holds. Therefore, quality indicators are introduced.

An unary quality indicator assigns a real valued quality to a set of solutions. Here, the hypervolume indicator (Zitzler and Thiele, 1998) and the ϵ-indicator (Zitzler et al., 2003) are measured. We use the performance assessment tools contributed to the PISA (Bleuler, Laumanns, Thiele, and Zitzler, 2003) software package with standard parameters.

The hypervolume indicator w.r.t. reference set A_{ref} is defined as

$$\mathcal{I}_{\mathcal{S},A_{\text{ref}}}(A) = \mathcal{S}_{a_{\text{nadir}}}(A_{\text{ref}}) - \mathcal{S}_{a_{\text{nadir}}}(A) \quad ,$$

where a_{nadir} denotes a reference point with the worst possible objective function values in each component. A smaller $\mathcal{I}_{\mathcal{S}}$ is preferable. The additive unary ϵ-indicator w.r.t. reference set A_{ref} is defined as

$$\mathcal{I}_{\epsilon,A_{\text{ref}}}(A) = \inf \left\{ \epsilon \in \mathbb{R} \,|\, \forall a \in A_{\text{ref}} \, \exists a' \in A \, \forall m \in \{1, \ldots, M\} : f_m(a) + \epsilon \geq f_m(a') \right\} \quad .$$

The ϵ-indicator determines the smallest offset by which the fitness values of the elements in A have to be shifted such that the resulting Pareto front covers the Pareto front of A_{ref} in the objective space. A smaller $\mathcal{I}_{\epsilon,A_{\text{ref}}}$ is preferable.

Before the performance indicators are computed, the data are normalized. We consider two slightly different ways of defining the reference sets. Assume we want to compare k algorithms on a particular optimization problem after a predefined number g of generations (this is the standard scenario when using the PISA software package). For each algorithm we have conducted t trials. We consider the non-dominated individuals of the union of all kt populations after g generations. Their objective vectors are normalized such that for every objective the smallest and largest objective function value are mapped to 1 and 2, respectively, by an affine transformation. These individuals make up the reference set A_{ref}. The mapping to $[1,2]^M$ is fixed and applied to all objective vectors under consideration. The reference point a_{ref} is chosen to have an objective value of 2.1 in each objective. Otherwise, if we want to compare the evolution of an indicator value over all generations, we consider the union of all populations over all algorithms, all trials, *and all generations* (i.e., $(G+1)kt$ populations if G is the number of the final generation) for normalization and computation of A_{ref} and proceed analogously.

Knowles and Corne (2002) and Zitzler et al. (2003) studied various properties of quality indicators. Of particular interest is the relation to the "being better" definition given above. An unary quality indicator is $\not\triangleright$-compatible, if a better indicator value for A than for B implies $B \not\triangleright A$. An indicator is $\triangleright$-complete, if $A \triangleright B$ implies a better indicator value for A than for B. Both the ϵ-indicator as well as the hypervolume indicator are $\not\triangleright$-compatible and $\triangleright$-complete. The crowding-distance measure described in section 3.2.2, which is related to the spread of solutions and not directly to the being better relation defined above, is neither $\not\triangleright$-compatible nor $\triangleright$-complete.

4.2 Experiments

Standard Benchmark Functions We consider three groups of test functions. The first group comprises six common benchmark problems taken from the literature, namely the function FON proposed by Fonseca and Fleming (1998) and the test functions ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 proposed by Zitzler, Deb, and Thiele (2000), see Table 4. All functions have box constraints also given in the table. In the five ZDT problems, most components of the optimal solution lie on the boundary of box constraints (which presumably favors NSGA-II). In general, we question the relevance of these test functions, because they are highly biased. Here, these problems are considered only because they are frequently used and we want to discover how the MO-CMA-ES compares to algorithms biased towards such kind of problems.

Unconstrained Test Functions with Quadratic Objectives The second group of benchmarks are functions where for each objective the objective function is quadratic (a quadratic approximation close to a local optimum is reasonable for any smooth enough fitness function), see Table 5. They are of the general form $f_m(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{O}_m^T \boldsymbol{A} \boldsymbol{O}_m \boldsymbol{x}$, where $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{Q}, \boldsymbol{O}_m, \boldsymbol{A} \in \mathbb{R}^{n \times n}$ with $\boldsymbol{O}_m$ orthogonal and $\boldsymbol{A}$ diagonal and positive definite. There are two types of functions, ELLI and CIGTAB, which differ in the eigenvectors of $\boldsymbol{Q}$. In each optimization run the coordinate system of the objective functions is changed by a random choice of $\boldsymbol{O}_m$. The *Gram-Schmidt orthogonalization procedure* can be used to construct random orthonormal basis vectors, the columns of $\boldsymbol{O}_m$, from randomly drawn

Table 4. Standard box constrained benchmark problems to be minimized.

Problem	n	Variable bounds	Objective functions	Optimal solution
FON	3	$[-4, 4]$	$f_1(\boldsymbol{x}) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i - \frac{1}{\sqrt{3}}\right)^2\right)$	$x_1 = x_2 = x_3$
			$f_2(\boldsymbol{x}) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i + \frac{1}{\sqrt{3}}\right)^2\right)$	$x_i \in [-1/\sqrt{3}, 1/\sqrt{3}]$
ZDT1	30	$[0, 1]$	$f_1(\boldsymbol{x}) = x_1$	$x_1 \in [0, 1]$
			$f_2(\boldsymbol{x}) = g(\boldsymbol{x})\left[1 - \sqrt{x_1/g(\boldsymbol{x})}\right]$	$x_i = 0$
			$g(\boldsymbol{x}) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	$i = 2, \dots n$
ZDT2	30	$[0, 1]$	$f_1(\boldsymbol{x}) = x_1$	$x_1 \in [0, 1]$
			$f_2(\boldsymbol{x}) = g(\boldsymbol{x})\left[1 - (x_1/g(\boldsymbol{x}))^2\right]$	$x_i = 0$
			$g(\boldsymbol{x}) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	$i = 2, \dots n$
ZDT3	30	$[0, 1]$	$f_1(\boldsymbol{x}) = x_1$	$x_1 \in [0, 1]$
			$f_2(\boldsymbol{x}) = g(\boldsymbol{x})\left[1 - \sqrt{x_1/g(\boldsymbol{x})} - \frac{x_1}{g(\boldsymbol{x})}\sin(10\pi x_1)\right]$	$x_i = 0$
			$g(\boldsymbol{x}) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	$i = 2, \dots n$
ZDT4	10	$x_1 \in [0, 1]$	$f_1(\boldsymbol{x}) = x_1$	$x_1 \in [0, 1]$
		$x_i \in [-5, 5]$	$f_2(\boldsymbol{x}) = g(\boldsymbol{x})\left[1 - \sqrt{x_1/g(\boldsymbol{x})}\right]$	$x_i = 0$
		$i = 2, \dots n$	$g(\boldsymbol{x}) = 1 + 10(n-1) + \sum_{i=2}^{n}\left[x_i^2 - 10\cos(4\pi x_i)\right]$	$i = 2, \dots n$
ZDT6	10	$[0, 1]$	$f_1(\boldsymbol{x}) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$	$x_1 \in [0, 1]$
			$f_2(\boldsymbol{x}) = g(\boldsymbol{x})\left[1 - (f_1(\boldsymbol{x})/g(\boldsymbol{x}))^2\right]$	$x_i = 0$
			$g(\boldsymbol{x}) = 1 + 9\left[\left(\sum_{i=2}^{n} x_i\right)/(n-1)\right]^{0.25}$	$i = 2, \dots n$

Table 5. Unconstrained benchmark problems to be minimized, 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.

Problem	n	Initial region	Objective functions	Optimal solution
ELLI_1	10	$[-10, 10]$	$f_1(\boldsymbol{y}) = \frac{1}{a^2 n}\sum_{i=1}^{n} a^{2\frac{i-1}{n-1}} y_i^2$	$y_1 = \dots = y_n$
			$f_2(\boldsymbol{y}) = \frac{1}{a^2 n}\sum_{i=1}^{n} a^{2\frac{i-1}{n-1}} (y_i - 2)^2$	$y_1 \in [0, 2]$
ELLI_2	10	$[-10, 10]$	$f_1(\boldsymbol{y}) = \frac{1}{a^2 n}\sum_{i=1}^{n} a^{2\frac{i-1}{n-1}} y_i^2$	
			$f_2(\boldsymbol{z}) = \frac{1}{a^2 n}\sum_{i=1}^{n} a^{2\frac{i-1}{n-1}} (z_i - 2)^2$	
CIGTAB_1	10	$[-10, 10]$	$f_1(\boldsymbol{y}) = \frac{1}{a^2 n}\left[y_1^2 + \sum_{i=2}^{n-1} a y_i^2 + a^2 y_n^2\right]$	$y_1 = \dots = y_n$
			$f_2(\boldsymbol{y}) = \frac{1}{a^2 n}\left[(y_1 - 2)^2 + \sum_{i=2}^{n-1} a\,(y_i - 2)^2 + a^2 (y_n - 2)^2\right]$	$y_1 \in [0, 2]$
CIGTAB_2	10	$[-10, 10]$	$f_1(\boldsymbol{y}) = \frac{1}{a^2 n}\left[y_1^2 + \sum_{i=2}^{n-1} a y_i^2 + a^2 y_n^2\right]$	
			$f_2(\boldsymbol{z}) = \frac{1}{a^2 n}\left[(z_1 - 2)^2 + \sum_{i=2}^{n-1} a\,(z_i - 2)^2 + a^2 (z_n - 2)^2\right]$	

vectors. In the case of the test functions ELLI_1 and CIGTAB_1 the same rotation $\boldsymbol{O}$ is used for both objective functions (i.e., $\boldsymbol{O}_1 = \boldsymbol{O}_2$). In the more general case of ELLI_2 and CIGTAB_2 two independent rotation matrices $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$ are generated, which are applied to the first and second objective function, respectively.

Table 6. New 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 $y_{\max} = 1/\max_j(|o_{1j}|)$. In the case of ZDT4', $o_{1j} = o_{j1} = 0$ for $1 < j \leq n$ and $o_{11} = 1$. For the definition of h, h_f, and h_g see Table 7.

Problem	n	Variable bounds	Objective function	Optimal solution
ZDT4'	10	$x_1 \in [0,1]$	$f_1(\boldsymbol{x}) = x_1$	$x_1 \in [0,1]$
		$x_i \in [-5,5]$	$f_2(\boldsymbol{x}) = g(\boldsymbol{y})\left[1 - \sqrt{x_1/g(\boldsymbol{y})}\right]$	$y_i = 0$
		$i = 2,\dots n$	$g(\boldsymbol{y}) = 1 + 10(n-1) + \sum_{i=2}^{n}\left[y_i^2 - 10\cos(4\pi y_i)\right]$	$i = 2,\dots n$
IHR1	10	$[-1,1]$	$f_1(\boldsymbol{x}) = \lvert y_1\rvert$	$y_1 \in [0, y_{\max}]$
			$f_2(\boldsymbol{x}) = g(\boldsymbol{y})\, h_f\!\left(1 - \sqrt{h(y_1)/g(\boldsymbol{y})}\right)$	$y_i = 0$
			$g(\boldsymbol{y}) = 1 + 9\left(\sum_{i=2}^{n} h_g(y_i)\right)/(n-1)$	$i = 2,\dots n$
IHR2	10	$[-1,1]$	$f_1(\boldsymbol{x}) = \lvert y_1\rvert$	$y_1 \in [-y_{\max}, y_{\max}]$
			$f_2(\boldsymbol{x}) = g(\boldsymbol{y})\, h_f\!\left(1 - (y_1/g(\boldsymbol{y}))^2\right)$	$y_i = 0$
			$g(\boldsymbol{y}) = 1 + 9\left(\sum_{i=2}^{n} h_g(y_i)\right)/(n-1)$	$i = 2,\dots n$
IHR3	10	$[-1,1]$	$f_1(\boldsymbol{x}) = \lvert y_1\rvert$	$y_1 \in [0, y_{\max}]$
			$f_2(\boldsymbol{x}) = g(\boldsymbol{y})\, h_f\!\left(1 - \sqrt{h(y_1)/g(\boldsymbol{y})} - \frac{h(y_1)}{g(\boldsymbol{y})}\sin(10\pi y_1)\right)$	$y_i = 0$
			$g(\boldsymbol{y}) = 1 + 9\left(\sum_{i=2}^{n} h_g(y_i)\right)/(n-1)$	$i = 2,\dots n$
IHR4	10	$[-5,5]$	$f_1(\boldsymbol{x}) = \lvert y_1\rvert$	$y_1 \in [0, y_{\max}]$
			$f_2(\boldsymbol{x}) = g(\boldsymbol{y})\, h_f\!\left(1 - \sqrt{h(y_1)/g(\boldsymbol{y})}\right)$	$y_i = 0$
			$g(\boldsymbol{y}) = 1 + 10(n-1) + \sum_{i=2}^{n}\left[y_i^2 - 10\cos(4\pi y_i)\right]$	$i = 2,\dots n$
IHR6	10	$[-1,1]$	$f_1(\boldsymbol{x}) = 1 - \exp(-4\lvert y_1\rvert)\sin^6(6\pi y_1)$	$y_1 \in [-y_{\max}, y_{\max}]$
			$f_2(\boldsymbol{x}) = g(\boldsymbol{y})\, h_f\!\left(1 - (f_1(\boldsymbol{x})/g(\boldsymbol{y}))^2\right)$	$y_i = 0$
			$g(\boldsymbol{y}) = 1 + 9\left[\left(\sum_{i=2}^{n} h_g(y_i)\right)/(n-1)\right]^{0.25}$	$i = 2,\dots n$

Table 7. Auxiliary functions for Table 6

$$
\begin{array}{lllll}
h: & \mathbb{R} \to [0,1], & x & \mapsto & \left(1 + \exp\left(\frac{-x}{\sqrt{n}}\right)\right)^{-1} \\
h_f: & \mathbb{R} \to \mathbb{R}, & x & \mapsto & \begin{cases} x & \text{if } \lvert y_1\rvert \leq y_{\max} \\ \lvert y_1\rvert + 1 & \text{otherwise} \end{cases} \\
h_g: & \mathbb{R} \to \mathbb{R}_{\geq 0}, & x & \mapsto & \frac{x^2}{\lvert x\rvert + 0.1}
\end{array}
$$

Generalized ZDT Problems The third group of problems shown in Table 6 are new benchmarks that generalize the ZDT problems to allow a rotation of the search space as in the second group. In the first function ZDT4' the rotation is applied to all but the first coordinates. That is, we consider $\boldsymbol{y} = \boldsymbol{O}\boldsymbol{x}$, where $\boldsymbol{O} \in \mathbb{R}^{n\times n}$ is an orthogonal matrix with $o_{1j} = o_{j1} = 0$ for $1 < j \leq n$ and $o_{11} = 1$.

In the other functions the rotation matrices are not restricted. Compared to the ZDT functions, the search space is expanded and the Pareto front is not completely located on the boundaries anymore. The lower end $y_1 = 0$ of the Pareto front is induced by the absolute value in the definition of f_1. The ends $y_1 = \pm y_{\max}$ of the Pareto front are

determined by h_f, see Table 7. The value $y_{\max}$ can be chosen between 1 and $1/\max_j(|o_{1j}|)$, and in the latter case the Pareto optimal solution $y_1 = y_{\max}$ lies on the search space boundary. If $y_{\max}$ is chosen larger, up to $\sum_j |o_{1j}|$ or $5\sum_j |o_{1j}|$, respectively, the Pareto front would not be linear in search space anymore. The function $h : \mathbb{R} \to [0, 1]$, see Table 7, is monotonic and emulates the original variable boundary $x_1 \in [0, 1]$. Similar, the function $h_g : \mathbb{R} \to \mathbb{R}_{\geq 0}$ emulates the original lower variable boundary of $x_{i \geq 0}$ for $i = 2, \ldots, n$.

NSGA-II We compare the c-MO-CMA and the s-MO-CMA with the real-coded non-dominated sorting genetic algorithm II (NSGA-II). The real-coded NSGA-II (Deb, 2001; Deb et al., 2002) uses non-dominated sorting and the crowding-distance for selection, and real-valued genetic algorithm (GA) operators, namely polynomial mutation and simulated binary crossover (SBX). A detailed description of how these operators work is given in Appendix A based on Deb and Agrawal (1999) and Deb et al. (2003). These operators have their roots in GAs and are tailored for box constraints. Note that they are particularly well-suited when the optimal solutions lie on the boundary of a box constraint. They operate component-wise and therefore implicitly favor separability. Thus, the NSGA-II is perfectly customized for the benchmark problems in Table 4.

NSDE To our knowledge, the only other evolutionary MOO approach that is invariant against rotation and rescaling of the search space is non-dominated sorting differential evolution (NSDE). Hence, we compare our methods to NSDE as described by Iorio and Li (2005).

In every generation g of NSDE, each parent $\boldsymbol{x}_i^{(g)}$ generates one offspring $\boldsymbol{x'}_i^{(g)}$ according to

$$\boldsymbol{x'}_i^{(g)} = \boldsymbol{x}_i^{(g)} + K\left(\boldsymbol{x}_{r_{i,3}^{(g)}}^{(g)} - \boldsymbol{x}_i^{(g)}\right) + F\left(\boldsymbol{x}_{r_{i,1}^{(g)}}^{(g)} - \boldsymbol{x}_{r_{i,2}^{(g)}}^{(g)}\right) \quad ,$$

where $r_{i,1}^{(g)}, r_{i,2}^{(g)}, r_{i,3}^{(g)} \in \{1, \ldots, \mu\}$ are randomly chosen indices obeying $\left|\{r_{i,1}^{(g)}, r_{i,2}^{(g)}, r_{i,3}^{(g)}, i\}\right| = 4$ and K and F are real-valued parameters. The new parents are selected from the former parents and their offspring by non-dominated sorting using the crowding-distance.

The described variation rule is known as *DE/current-to-rand/1* in single-objective differential evolution (Price, 1999). The individuals in the population span the subspace of the search space reachable by the algorithm. All offspring are linearly dependent from the parents. This bears the risk that selection may lead to a degenerated population that is restricted to some subspace not containing the desired solutions. The risk depends on the relation between the dimension n of the search space and the population size. The higher the dimension and the smaller the population size the higher is this risk.

Parameter Setting and Initialization For the real-coded NSGA-II we used the same parameter setting as Deb et al. (2002). We set the mutation probability to the inverse of the genotype space dimension, $p_m = n^{-1}$, and the crossover probability to $p_c = 0.9$. The distribution indices of the crossover and mutation operator were set to $\eta_c = \eta_m = 20$. In the case of the unconstrained benchmark functions in Table 5 the boundaries of the mutation and crossover operator were set to the boundaries of the initial regions. See Appendix A for a description of the real-coded NSGA-II variation operators and their parameters.

The parameters of the NSDE were set to $K = 0.4$ and $F = 0.8$ as done by Iorio and Li (2005). Constraints are handled as in the evolution strategies.

We used the standard parameters of the (1+1)-CMA-ES in the MO-CMA-ES. For the functions FON, ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 we set $\sigma^{(0)}$ equal to 60 %

Table 8. Results on common benchmark problems. The upper two and lower two tables show the median of 100 trials after 50000 evaluations of the hypervolume-indicator and the ϵ-indicator, respectively. The smallest value in each column is underlined, the largest is printed in italics. The superscripts I, II, III, and IV indicate whether an algorithm is statistically significantly better than the c-MO-CMA, s-MO-CMA, NSGA-II, and NSDE, respectively (two-sided Wilcoxon rank sum test, $p < 0.001$).

hypervolume indicator			
algorithm	FON	ZDT1	ZDT2
s-MO-CMA	$\underline{0.00467}^{\mathsf{II},\mathsf{III},\mathsf{IV}}$	$\underline{0.00217}^{\mathsf{II},\mathsf{III},\mathsf{IV}}$	$\underline{0.00247}^{\mathsf{II},\mathsf{III},\mathsf{IV}}$
c-MO-CMA	$0.00643^{\mathsf{III},\mathsf{IV}}$	0.00375^{IV}	0.00416^{IV}
NSGA-II	*0.00855*	$0.00264^{\mathsf{II},\mathsf{IV}}$	0.00316^{IV}
NSDE	0.00719^{III}	*0.10872*	*0.09133*

hypervolume indicator			
algorithm	ZDT3	ZDT4	ZDT6
s-MO-CMA	$\underline{0.00105}^{\mathsf{II},\mathsf{III},\mathsf{IV}}$	0.22792^{IV}	$\underline{0.00051}^{\mathsf{II},\mathsf{III},\mathsf{IV}}$
c-MO-CMA	0.00186^{IV}	0.22286^{IV}	0.00064^{IV}
NSGA-II	$0.00140^{\mathsf{II},\mathsf{IV}}$	$\underline{0.00016}^{\mathsf{I},\mathsf{II},\mathsf{IV}}$	$0.00062^{\mathsf{II},\mathsf{IV}}$
NSDE	*0.09326*	*0.80156*	*0.00121*

ϵ-indicator			
algorithm	FON	ZDT1	ZDT2
s-MO-CMA	$\underline{0.00684}^{\mathsf{II},\mathsf{III},\mathsf{IV}}$	$\underline{0.00459}^{\mathsf{II},\mathsf{III},\mathsf{IV}}$	$\underline{0.00502}^{\mathsf{II},\mathsf{III},\mathsf{IV}}$
c-MO-CMA	0.01414	0.01124^{IV}	0.01280^{IV}
NSGA-II	0.01388	$0.00818^{\mathsf{II},\mathsf{IV}}$	0.01033
NSDE	*0.01436*	*0.08017*	*0.08533*

ϵ-indicator			
algorithm	ZDT3	ZDT4	ZDT6
s-MO-CMA	$\underline{0.00317}^{\mathsf{II},\mathsf{III},\mathsf{IV}}$	0.21138^{IV}	$\underline{0.00148}^{\mathsf{II},\mathsf{III},\mathsf{IV}}$
c-MO-CMA	0.00870^{IV}	0.20985^{IV}	0.00305
NSGA-II	$0.00711^{\mathsf{II},\mathsf{IV}}$	$\underline{0.00186}^{\mathsf{I},\mathsf{II},\mathsf{IV}}$	$0.00256^{\mathsf{II},\mathsf{IV}}$
NSDE	*0.09936*	*0.73511*	*0.00328*

of $x_2^u - x_2^l$ (we rescaled the first component of ZDT4 to $[-5, 5]$). In the unconstrained problems, Table 5, we set $\sigma^{(0)}$ equal to 60 % of the initialization range of one component.

In all algorithms the population size (λ_{MO}) was set to 100 as in the study by Deb et al. (2002) to allow for a better comparison.

Methods For each pair of test function and optimization algorithm 100 trials with different initial populations were conducted. For each test problem, the 100 initial populations and the randomly generated rotation matrices for the rotated problems were the same for each algorithm.

Results The characteristics of the Pareto fronts after 500 generations (50000 fitness evaluations) are shown in Table 8, Table 9, and Table 10 for the three groups of benchmark problems. The superscripts I, II, III, and IV indicate whether a value is statistically significantly compared to the c-MO-CMA, s-MO-CMA, NSGA-II, and NSDE, respectively (paired Wilcoxon rank sum test, $p < 0.001$, superscripts in italics refer to a significance

Table 9. Results on new unconstrained, rotated benchmark problems. The upper two and lower two tables show the median of 100 trials after 50000 evaluations of the hypervolume-indicator and the ϵ-indicator, respectively.

hypervolume indicator				
algorithm	$ELLI_1$	$ELLI_2$	$CIGTAB_1$	$CIGTAB_2$
s-MO-CMA	$\underline{0.00345}^{II,III,IV}$	0.00003^{III}	$\underline{0.00314}^{II,III,IV}$	$0.00001^{III,IV}$
c-MO-CMA	$0.00624^{III,IV}$	0.00003^{III}	$0.00545^{III,IV}$	$\underline{0.00000}^{I,III,IV}$
NSGA-II	*0.00750*	*0.00023*	0.00584^{IV}	*0.00005*
NSDE	0.00687^{III}	$\underline{0.00002}^{I,II,III}$	*0.00694*	0.00001^{III}

ϵ-indicator				
algorithm	$ELLI_1$	$ELLI_2$	$CIGTAB_1$	$CIGTAB_2$
s-MO-CMA	$\underline{0.00577}^{II,III,IV}$	$0.00011^{II,III}$	$\underline{0.00561}^{II,III,IV}$	$0.00019^{II,III}$
c-MO-CMA	0.01378	0.00013^{III}	0.01357	0.00022^{III}
NSGA-II	0.01305^{IV}	*0.00049*	*0.01418*	*0.00033*
NSDE	*0.01405*	$\underline{0.00009}^{I,II,III}$	0.01405	$\underline{0.00018}^{I,II,III}$

Table 10. Results on new, rotated, constrained benchmark problems. The upper and lower table show the median of 100 trials after 50000 evaluations of the hypervolume-indicator and the ϵ-indicator, respectively.

hypervolume indicator			
algorithm	ZDT4'	IHR1	IHR2
s-MO-CMA	$\underline{0.16774}^{IV}$	$0.00323^{III,IV}$	$\underline{0.04140}^{II,III,IV}$
c-MO-CMA	0.18962^{IV}	$\underline{0.00284}^{III,IV}$	0.04323^{III}
NSGA-II	0.18282^{IV}	0.01939^{IV}	*0.06383*
NSDE	*0.75090*	*0.02012*	$0.04289^{II,III}$

hypervolume indicator			
algorithm	IHR3	IHR4	IHR6
s-MO-CMA	$\underline{0.02401}^{II,III,IV}$	$\underline{0.00683}^{III,IV}$	$0.01093^{III,IV}$
c-MO-CMA	$0.02402^{III,IV}$	$0.00759^{III,IV}$	$\underline{0.01076}^{III,IV}$
NSGA-II	0.02409^{IV}	0.01725^{IV}	*0.04053*
NSDE	*0.02415*	*0.03600*	0.02391^{III}

ϵ-indicator			
algorithm	ZDT4'	IHR1	IHR2
s-MO-CMA	0.16626^{IV}	$0.01053^{III,IV}$	$\underline{0.16396}^{III,IV}$
c-MO-CMA	0.18465^{IV}	$\underline{0.00937}^{III,IV}$	$0.16428^{III,IV}$
NSGA-II	$\underline{0.16531}^{IV}$	0.03147^{IV}	*0.21648*
NSDE	*0.69407*	*0.03214*	0.16497^{III}

ϵ-indicator			
algorithm	IHR3	IHR4	IHR6
s-MO-CMA	$\underline{0.03996}^{III,IV}$	$\underline{0.00669}^{III,IV}$	$\underline{0.02123}^{III}$
c-MO-CMA	$0.03996^{III,IV}$	$0.00746^{III,IV}$	0.02170^{III}
NSGA-II	0.04003^{IV}	0.01777^{IV}	*0.05727*
NSDE	*0.04008*	*0.03321*	0.02899^{III}

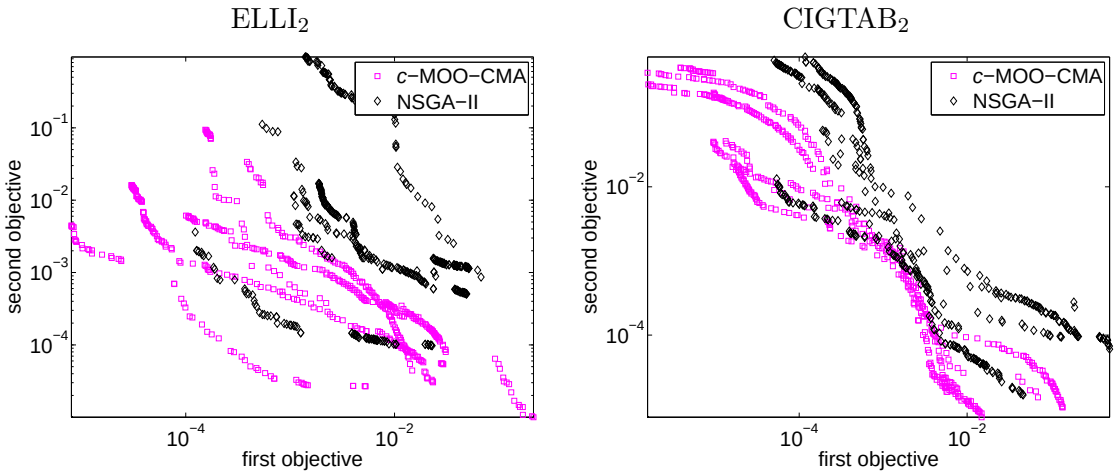

Figure 2. Population plots in objective space for c-MO-CMA and NSGA-II on the test functions ELLI_2 and CIGTAB_2 on logarithmic scale. The populations after 500 generations of the first 5 trials are shown. Note the different shapes of the Pareto fronts due to the different coordinate transformations and that s-MO-CMA, which is not shown in this figure, has a significantly better performance on these functions even compared to c-MO-CMA.

Table 11. Results on the new benchmark problems ELLI_2, CIGTAB_2, and ZDT4' after 100000 evaluations. The upper and lower table show the median of 100 trials of the hypervolume-indicator and the ϵ-indicator, respectively. In the corresponding figure, the populations after 1000 generations of the first 5 trials on ZDT4' are plotted.

hypervolume indicator			
algorithm	ELLI_2	CIGTAB_2	ZDT4'
s-MO-CMA	$0.00001^{\text{III,IV}}$	$\underline{0.00000}^{\text{II,III,IV}}$	$\underline{0.15583}^{\text{III,IV}}$
c-MO-CMA	$\underline{0.00001}^{\text{I,III,IV}}$	$0.00000^{\text{III,IV}}$	$0.18808^{\text{III,IV}}$
NSGA-II	*0.00018*	*0.00005*	0.22316^{IV}
NSDE	0.00002^{III}	0.00001^{III}	*0.80157*

ϵ-indicator			
algorithm	ELLI_2	CIGTAB_2	ZDT4'
s-MO-CMA	$\underline{0.00009}^{\text{II,III,IV}}$	$\underline{0.00033}^{\text{II,III,IV}}$	$\underline{0.14434}^{\text{III,IV}}$
c-MO-CMA	0.00014^{III}	0.00042^{III}	$0.17247^{\text{III,IV}}$
NSGA-II	*0.00044*	*0.00073*	0.20273^{IV}
NSDE	$0.00010^{\text{II,III}}$	$0.00034^{\text{II,III}}$	*0.73926*

level of $p < 0.01$). Figure 2 shows population plots of the first five trials after 500 generations of the c-MO-CMA and NSGA-II for ELLI_2 and CIGTAB_2 in the objective space. Performance indicators for ELLI_2, CIGTAB_2, and ZDT4' after 1000 generations are given in Table 11. A corresponding population plot for c-MO-CMA and NSGA-II on ZDT4' after 1000 is shown in Figure 3. In Figure 4, the evolution of the median of the ϵ-Indicator is shown for the four test problems with quadratic objective functions. As described in Section 4.1, the reference sets and therefore the absolute values in Figure 4 are different from those in Table 9, although they are computed from the same trials.

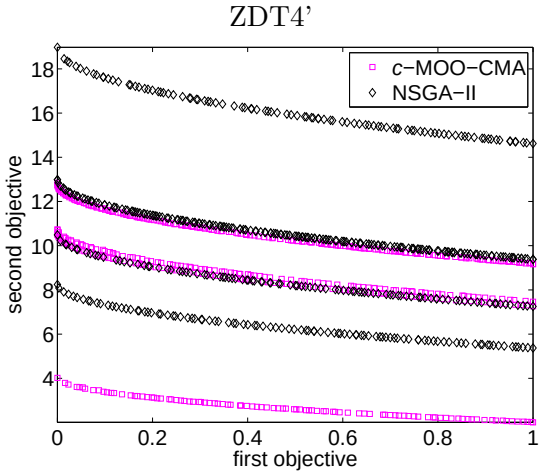

Figure 3. Populations generated by c-MO-CMA and NSGA-II after 1000 generations on ZDT4'. The first 5 of the 100 trials described in Table 11 are plotted.

Discussion The three methods NSGA-II, NSDE, and c-MO-CMA rely on the same multi-objective selection pressure. Comparing their performance in terms of the ϵ-indicator and the hypervolume indicator allows for a fair comparison of the different variation strategies. Because the selection in s-MO-CMA almost directly aims at optimizing hypervolume, using the latter for comparisons of s-MO-CMA with the other algorithms is biased.

When looking at both the ϵ-indicator and the hypervolume indicator in the tables, s-MO-CMA is statistically significantly better than NSGA-II in all benchmark problems except ZDT4, where NSGA-II is significantly better, and ZDT4', where after 500 generations the lower figures of the evolution strategies are not statistically significant (see below).

The multi-modal ZDT4 is separable, in the sense that the optima form a regular axis-parallel grid. The recombination operator in NSGA-II exploits this kind of separability by combining (locally) optimal settings of different variables: the crossover of local optima always delivers another optimum being better in almost half of the cases and eventually the global optimum. MO-CMA-ES does not exploit this kind of separability and cannot find close to optimal solutions. When the search space is rotated such that the optima do not necessarily lie on a regular axis-parallel grid, NSGA-II is not superior anymore as can be seen from the ZDT4' results. On the contrary, the evolution strategies become better. After 500 generations the differences are not significant at $p < 0.01$, because the median absolute deviation (and the variance) for the single algorithms is quite high due to the huge number of local optima of the ZDT4' function. However, after 1000 generations the evolution strategies are significantly better than NSGA-II w.r.t. both performance indicators, see Table 11. Figure 3 shows that on ZDT4' NSGA-II suffers more under premature convergence than the evolution strategies.

The s-MO-CMA differs in two main aspects from the NSGA-II. First, the adaptation of the individual Gaussian mutation distributions by the CMA instead of using real-valued GA operators. Second, the sorting is based on the contributing hypervolume instead of the crowding-distance. To investigate the impact of these two differences, we compare c-MO-CMA and s-MO-CMA, which differ only in the selection scheme, and c-MO-CMA and NSGA-II. The latter two algorithms differ in the variation operators, but have the

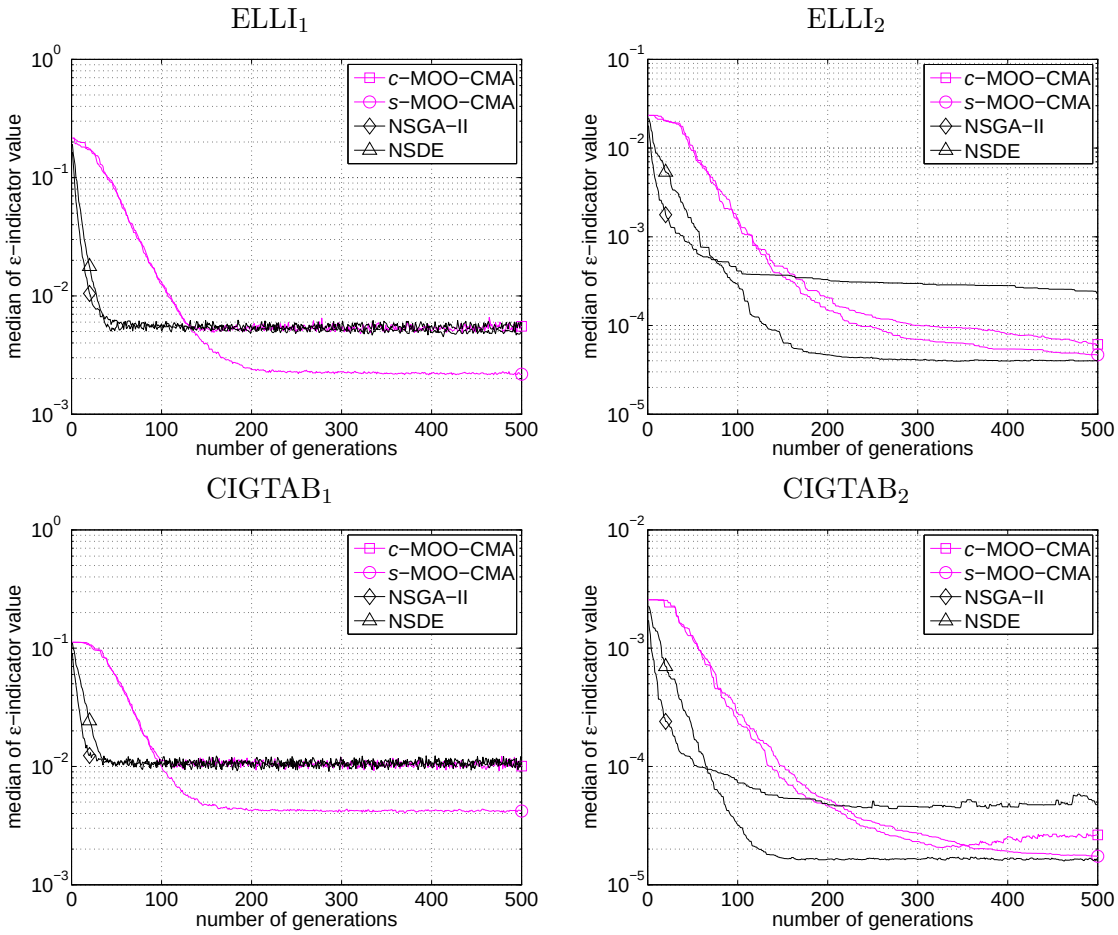

Figure 4. Simulations of the multi-objective algorithms on four rotated test functions. Shown is the median of the ϵ-indicator of the first 20 trials. The reference sets and therefore the absolute values are different compared to Table 9, see Section 4.1. In the first approximately 100 and 120 generations, the plots for c-MO-CMA and s-MO-CMA overlap in case of ELLI_1 and CIGTAB_1, respectively. After that, the plots for c-MO-CMA, NSGA-II, and NSDE can hardly be distinguished. On CIGTAB_2 and in particular on ELLI_2, the plots for c-MO-CMA and s-MO-CMA are very close. Note that ELLI_2 and CIGTAB_2 are the only benchmark problems considered in this study where the NSDE outperforms the other methods.

same selection mechanism.

The ZDT experiments clearly show that the selection based on the hypervolume leads to better results in terms of the measured indicators, because here s-MO-CMA significantly outperforms c-MO-CMA.

On the common benchmark problems, Table 4, the NSGA-II is superior to the c-MO-CMA. The ϵ-indicator values are significantly better on ZDT1, ZDT4, and ZDT6, the hypervolume indicator values additionally on ZDT3. On these functions, NSGA-II can take advantage of the separability as described above. On FON, the c-MO-CMA is significantly better than the NSGA-II in terms of the hypervolume indicator.

On the rotated benchmark problems with one global coordinate system, ELLI_1 and CIGTAB_1, the c-MO-CMA and the NSGA-II do not differ significantly w.r.t ϵ-indicator but only w.r.t. the hypervolume indicator. The reason why the c-MO-CMA does not reach

better ϵ-indicator values than the NSGA-II—despite the fact that the evolution strategy can adapt its mutation distribution to the rotated coordinate systems—lies in the selection mechanism. After an initial phase, NSGA-II and c-MO-CMA as well as NSDE suffer from the fact that the crowding-distance is not related to the being better relation defined in Section 4.1. Depending on the population size, this limits the progress of the algorithms in terms of the ϵ-indicator (and, although not obvious from all presented results, also in terms of the hypervolume indicator). This can be observed in Figure 4, left. After approximately 120 and 100 generations on ELLI_1 and CIGTAB_1, respectively, the three methods relying on the crowding-distance fluctuate around a sub-optimal level without any progress w.r.t. the ϵ-indicator. Their final performance is determined by the second sorting criterion, the corresponding plots can not be distinguished.

When looking at the problems ELLI_2 and CIGTAB_2, where, roughly speaking, the appropriate coordinate system varies along the Pareto front, both MO-CMA-ES variants clearly outperform NSGA-II. In this general case, the adaptation of arbitrary normal mutation distributions, individually along the Pareto front, seems to be of great importance. The resulting Pareto fronts are visualized in Figure 2 (note that the s-MO-CMA leads to even better results on these functions). On the IHR problems, c-MO-CMA is significantly better than NSGA-II w.r.t. both indicators. These results confirm that the invariance properties of the MO-CMA really matter, see also Figure 4, right.

Figure 4 reveals that the evolution strategies are slower in the early generations on the four test problems with quadratic objective functions compared to the other methods. It takes some time for the CMA to adapt the strategy parameters during which NSDE and NSGA-II make significantly more progress (at least for the initial CMA strategy parameters and learning rates used in this study).

Overall the NSDE performed worst of all methods. However, on ELLI_2 and CIGTAB_2, until 500 generations most indicator values are better than the corresponding values of the evolution strategies. This changes after more evaluations, see Table 11. This result does not carry over to the other benchmarks with rotated search spaces. On some of the IHR tasks, IHR1, IHR3, and IHR4, NSDE is even significantly worse than the NSGA-II. The differential evolution algorithm in the considered form seems to have problems with higher dimensionality, as can be seen from the results on ZDT1, ZDT2, and ZDT3, as well as multi-modality, as reflected by the performance on ZDT4 and ZDT4'.

The question arises whether tuning of the external parameters of the MO-CMA-ES, NSDE, or NSGA-II would qualitatively affect their performance. We conducted a parameter study for the NSGA-II with different values for p_m, p_c, η_m, and η_c on the new benchmark problems with quadratic objective functions without observing remarkably improved behavior. We think that the experiments in this section well reflect principal theoretical advantages and limitations of the algorithms.

5 Summary and Conclusions

We presented the single-objective $(1+\lambda)$-CMA-ES, an elitist evolution strategy (ES) with covariance matrix adaptation (CMA). It combines plus-selection and success rule based step size control with the powerful covariance matrix adaptation. The empirical evaluation reveals that the (1+1)-CMA-ES works reliably and that it is faster by a factor of about 1.5 on unimodal functions compared to the standard CMA-ES with comma-selection. The result proves that non-elitist (comma-) selection is *not* a necessary prerequisite for complex strategy parameter adaptation in evolution strategies.

While the new (1+1)-CMA-ES is slightly faster than the default $(\mu/\mu, \lambda)$-CMA-ES, it is more susceptible to get trapped into sub-optimal local minima. In particular for this reason we stick to the comma-variant as default recommendation for single-objective optimization.

Based on the (1+λ)-CMA-ES we developed the $\lambda_{\mathrm{MO}}\times$(1+$\lambda$)-MO-CMA-ES, a multi-objective CMA-ES, which combines the strategy parameter adaptation of λ_{MO} elitist (1+λ) strategies with multi-objective selection based on non-dominated sorting. Two variants were considered, *c*-MO-CMA and *s*-MO-CMA, using the crowding-distance and the contributing hypervolume as second sorting criterion, respectively.

The MO-CMA strategies are independent of the chosen coordinate system. Apart from the respective initializations, their behavior does not change if the search space is translated, rotated, and/or rescaled. The single-objective CMA-ES with plus-selection is additionally invariant against order-preserving transformations of the fitness function value, the MO-CMA-ES is not, because of the second level sorting criterion for selection. However, in comparison to other multi-objective evolutionary algorithms, the invariance properties of the MO-CMA-ES are an important feature.

In experiments we compared *c*-MO-CMA, *s*-MO-CMA, NSGA-II, and the differential evolution approach NSDE. The *s*-MO-CMA algorithm appears to be the superior method. It significantly outperforms the NSGA-II on all but one of the considered test problems. The NSGA-II is faster than the *s*-MO-CMA on problems where the optima form a regular axis-parallel grid, because NSGA-II heavily exploits this kind of separability. However, otherwise *s*-MO-CMA is superior. This clearly shows that both the new selection mechanism and in particular the invariance properties due to covariance matrix adaptation improve the search behavior in case of the *s*-MO-CMA. The rotation-invariant NSDE showed the overall worst performance of all methods, especially in higher dimensional problems, but gave good results on the two test problems where the appropriate coordinate system varies along the Pareto front.

The ranking in the *s*-MO-CMA, based on the contributing hypervolume, can be computed in superlinear time in the number of individuals for two objectives, but the algorithm scales badly for an increasing number of goals. We do not regard the bad scaling behavior as a severe drawback, in particular because in multi-objective optimization applications usually less than five objectives are considered. This is not only because the applications do not give raise to more objectives, but also because otherwise the results would be too hard to interpret (e.g., to visualize). Further, in real-world applications the costs for generating offspring and selection can often be neglected compared to the time needed for the fitness evaluation. If the contributing hypervolume cannot be used for selection because of a high number of objectives, the *c*-MO-CMA provides an alternative.

In conclusion, with the caveat of the so far limited empirical data basis, the *s*-MO-CMA is a promising candidate to become the method of choice for real-valued non-separable optimization problems with multiple criteria given that the maximum number of fitness evaluations is not too small to allow for an adaptation of the strategy parameters.

Acknowledgments

We thank K. Deb and co-workers and C. M. Fonseca, J. D. Knowles, L. Thiele, and E. Zitzler for making their software available. The first author gratefully acknowledges support from the Honda Research Institute Europe GmbH.

A NSGA-II Operators

The NSGA-II uses the polynomial mutation operator for optimization problems with box constrains (Deb and Agrawal, 1999; Deb et al., 2003). Let $\boldsymbol{c} = (c_1, \ldots, c_n)$ with $c_i \in [x_i^l, x_i^u], 1 \leq i \leq n$. The parameter $\eta_m > 0$ is called the distribution index of the mutation.

Procedure `mutatePolynomial`($\boldsymbol{c} \in [x_1^l, x_1^u] \times \cdots \times [x_n^l, x_n^u]$)

foreach $1 \leq i \leq n$ **do**
  $u \rightsquigarrow U[0,1]$
  if $u \leq p_m$ **then**
    $\alpha \leftarrow \dfrac{\min\{c_i - x_i^l, x_i^u - c_i\}}{(x_i^u - x_i^l)}$
    $z \rightsquigarrow U[0,1]$
    $\delta \leftarrow \begin{cases} [(2z) + (1-2z)(1-\alpha)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} - 1 & \text{, if } z \leq 0.5 \\ 1 - [2\cdot(1-z) + 2\cdot(z-0.5)(1-\alpha)^{\eta_m+1}]^{\frac{1}{\eta_m+1}} & \text{, otherwise} \end{cases}$
    $c_i \leftarrow c_i + \delta \cdot (x_i^u - x_i^l)$

The simulated binary crossover operator (`SBX`) for constrained problems (Deb and Agrawal, 1999; Deb et al., 2003) with distribution index $\eta_c > 0$ is defined as follows.

Procedure `SBX`($\boldsymbol{c_1}$,$\boldsymbol{c_2} \in [x_1^l, x_1^u] \times \cdots \times [x_n^l, x_n^u]$)

foreach $1 \leq i \leq n$ **do**
  $u \rightsquigarrow U[0,1[$
  if $u \geq 0.5$ **then**
    $y_1 \leftarrow \min(c_{1i}, c_{2i})$
    $y_2 \leftarrow \max(c_{1i}, c_{2i})$
    if $(y_2 - y_1) > \epsilon$ **then**
      $\beta \leftarrow 1 + \dfrac{2}{y_2 - y_1} \cdot \min\{(y_1 - x_i^l), (x_i^u - y_2)\}$
      $\alpha \leftarrow 2 - \beta^{-(\eta_c+1)}$
      $z \rightsquigarrow U[0,1]$
      $\gamma \leftarrow \begin{cases} (z\gamma)^{\frac{1}{\eta_c+1}} & \text{, if } z \leq \frac{1}{\alpha} \\ \left(\frac{1}{2-z\gamma}\right)^{\frac{1}{\eta_c+1}} & \text{, otherwise} \end{cases}$
    else
      $\gamma \leftarrow 1$
    $[\boldsymbol{c}_1]_i \leftarrow 0.5 \cdot [(y_1 + y_2) - \gamma \cdot (y_2 - y_1)]$
    $[\boldsymbol{c}_2]_i \leftarrow 0.5 \cdot [(y_1 + y_2) + \gamma \cdot (y_2 - y_1)]$

The parameter ϵ, which determines when two values are regarded as too close, is set to $\epsilon = 10^{-12}$. Due to numerical problems, these operators rather frequently hit the upper and lower bounds. In these cases, the mutation operator sets the corresponding variable x_i to some value chosen from $[x_i^l, x_i^u]$ uniformly at random.

References

Beyer, H.-G. and H.-P. Schwefel (2002). Evolution strategies: A comprehensive introduction. *Natural Computing 1*(1), 3–52.

Bleuler, S., M. Laumanns, L. Thiele, and E. Zitzler (2003). PISA – A platform and programming language independent interface for search algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele (Eds.), *Evolutionary Multi-Criterion Optimization (EMO 2003)*, Volume 2632 of *LNCS*, pp. 494 – 508. Springer-Verlag.

Büche, D., S. D. Müller, and P. Koumoutsakos (2003). Self-adaptation for multi-objective evolutionary algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele (Eds.), *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, Volume 2632 of *LNCS*, pp. 267 – 281. Springer-Verlag.

Coello Coello, C. A., D. A. Van Veldhuizen, and G. B. Lamont (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers.

Das, I. and J. E. Dennis (1997). A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization 14*(1), 63–69.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley.

Deb, K. et al. (2003).
http://www.iitk.ac.in/kangal/code/new_nsga/nsga2code.tar.

Deb, K. and S. Agrawal (1999). A niched-penalty approach for constraint handling in genetic algorithms. In R. Albrecht, A. Dobnikar, D. Pearson, and N. Steele (Eds.), *International Conference on Artifical Neural Networks and Genetic Algorithms*, pp. 235–243. Springer-Verlag.

Deb, K., S. Agrawal, A. Pratap, and T. Meyarivan (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation 6*(2), 182–197.

Emmerich, M., N. Beume, and B. Naujoks (2005). An EMO algorithm using the hypervolume measure as selection criterion. In C. A. C. Coello, E. Zitzler, and A. H. Aguirre (Eds.), *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, Volume 3410 of *LNCS*, pp. 62–76. Springer-Verlag.

Fonseca, C. M. and P. J. Fleming (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms—Part II: Application example. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 28*(1), 38–47.

Fonseca, C. M., J. D. Knowles, L. Thiele, and E. Zitzler (2005). A tutorial on the performance assessment of stochastic multiobjective optimizers. Presented at the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005).

Hansen, N. (2000). Invariance, self-adaptation and correlated mutations in evolution strategies. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN VI)*, Volume 1917 of *LNCS*, pp. 355–364. Springer-Verlag.

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

Hansen, N. (2005b). References to CMA-ES applications.
www.bionik.tu-berlin.de/user/niko/cmaapplications.pdf.

Hansen, N. (2006). An analysis of mutative σ-self-adaptation on linear fitness functions. *Evolutionary Computation*, accepted.

Hansen, N. and S. Kern (2004). Evaluating the CMA evolution strategy on multimodal test functions. In X. Yao et al. (Eds.), *Parallel Problem Solving from Nature - PPSN VIII, LNCS 3242*, pp. 282–291. Springer-Verlag.

Hansen, N., S. D. Müller, and P. Koumoutsakos (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adapt ation (CMA-ES). *Evolutionary Computation 11*(1), 1–18.

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

Hansen, N., A. Ostermeier, and A. Gawelczyk (1995). On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In L. Eshelman (Ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*, Pittsburgh, pp. 57–64. Morgan Kaufmann, San Fransisco.

Igel, C. (2005). Multi-objective model selection for support vector machines. In C. A. C. Coello, E. Zitzler, and A. H. Aguirre (Eds.), *Third International Conference on Evolutionary Multi-Criterion Op timization (EMO 2005)*, Volume 3410 of *LNAI*, pp. 534–546. Springer-Verlag.

Iorio, A. and X. Li (2005). Solving rotated multi-objective optimization problems using differential evolution. In *Proceeding of the 17th Joint Australian Conference on Artificial Intelligence*, LNCS. Spriner-Verlag. In press.

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

Knowles, J., L. Thiele, and E. Zitzler (2005, July). A tutorial on the performance assessment of stochastic multiobjective optimizers. 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.

Knowles, J. D. and D. W. Corne (2002). On metrics for comparing non-dominated sets. In *Congress on Evolutionary Computation Conference (CEC 2002)*, pp. 711–716. IEEE Press.

Knuth, D. E. (1973). *The art of computer programming* (1 ed.), Volume 3: Sorting and searching, Chapter 6, pp. 451–471. Addison-Wesley.

Laumanns, M., G. Rudolph, and H.-P. Schwefel (2001). Mutation control and convergence in evolutionary multi-objective optimization. In R. Matousek and P. Osmera (Eds.), *Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001)*, pp. 24–29. Brno, Czech Republic: University of Technology.

Price, K. V. (1999). An introduction to differential evolution. In D. Corne, M. Dorigo, and F. Glover (Eds.), *New Ideas in Optimization*, London, pp. 79–108. McGraw-Hill.

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog.

Salomon, R. (1996). Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions. *BioSystems 39*(3), 263–278.

Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology Series. John Wiley & Sons.

While, L. (2005). A new analysis of the LebMeasure algorithm for calculating hypervolume. In C. A. C. Coello, E. Zitzler, and A. H. Aguirre (Eds.), *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, Volume 3410 of *LNCS*, pp. 326–340. Springer-Verlag.

Zitzler, E., K. Deb, and L. Thiele (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation 8*(2), 173–195.

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

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