Steady-state Selection and Efficient Covariance Matrix Update in the Multi-objective CMA-ES

Christian Igel[1], Thorsten Suttorp[1], and Nikolaus Hansen[2]

[1] Institut für Neuroinformatik, Ruhr-Universität Bochum, 44780 Bochum, Germany
{christian.igel, thorsten.suttorp}@neuroinformatik.rub.de
[2] Institute of Computational Science, ETH Zurich, 8092 Zurich, Switzerland
nikolaus.hansen@inf.ethz.ch

Abstract. The multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES) combines a mutation operator that adapts its search distribution to the underlying optimization problem with multi-criteria selection. Here, a generational and two steady-state selection schemes for the MO-CMA-ES are compared. Further, a recently proposed method for computationally efficient adaptation of the search distribution is evaluated in the context of the MO-CMA-ES.

1 Introduction

Evolution strategies (ES) for real-valued optimization rely on Gaussian random variations. Appropriately adapting the covariance matrices of these mutations during optimization allows for learning a variable metric for the search distribution. It is well known that such an automatic adaptation of the mutation distribution drastically improves the search performance on non-separable and/or badly scaled single-objective functions [1–4].

In [5], we incorporated the step size and covariance matrix adaptation from the covariance matrix adaptation ES (CMA-ES, [3]) into a multi-objective framework. The resulting MO-CMA-ES used generational selection based on [6] combined with the sorting criterion proposed in [7, 8]. We chose generational selection in order to make our performance comparisons with alternative methods easier to interpret. However, in [7, 8] steady-state selection is used with good results and the question arises whether the MO-CMA-ES would profit from this selection scheme. In [9], we presented a new, computationally efficient update scheme for covariance matrices. The complexity reduction from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ per update of the mutation distribution, where n is the dimensionality of the search space, comes at the cost of slower adaptation rates. However, as in the MO-CMA-ES many mutation distributions need to be traced, this approach seems to be particularly promising for the MO-CMA-ES.

In this work, we first investigate the computationally efficient update proposed in [9] within the framework of the MO-CMA-ES. Second, we compare variants of the MO-CMA-ES with different steady-state selection schemes and generational selection, respectively.

2 Covariance Matrix Adaptation

Let us consider an additive mutation $\boldsymbol{v}_i^{(g)} \in \mathbb{R}^n$ of individual i in generation g. The mutation $\boldsymbol{v}_i^{(g)}$ is a realization of an n-dimensional random vector distributed according to a zero-mean Gaussian distribution with covariance matrix $\boldsymbol{C}_i^{(g)}$, that is, $\boldsymbol{v}_i^{(g)} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{C}_i^{(g)}\right)$. To sample this mutation distribution, n independent standard normally distributed random numbers are drawn to generate a realization of an n-dimensional normally distributed random vector $\boldsymbol{z}_i^{(g)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ with unit covariance matrix and zero mean. Then this random vector is rotated and scaled by a linear transformation $\boldsymbol{A}_i^{(g)} \in \mathbb{R}^{n \times n}$ such that

$$\boldsymbol{A}_i^{(g)} \boldsymbol{z}_i^{(g)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C}_i^{(g)}) \text{ for } \boldsymbol{z}_i^{(g)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \ .$$

Thus, for sampling the mutation distribution the covariance matrix $\boldsymbol{C}_i^{(g)}$ has to be decomposed into Cholesky factors $\boldsymbol{C}_i^{(g)} = \boldsymbol{A}_i^{(g)} {\boldsymbol{A}_i^{(g)}}^T$. One of the decisive features of ES is that the covariance matrices are subject to adaptation. The general policy is to alter the covariance matrices such that steps promising larger fitness gain are sampled more often. Here we consider matrix updates of the form $\boldsymbol{C}^{(g+1)} = \alpha \boldsymbol{C}^{(g)} + \beta \boldsymbol{V}^{(g)}$, where $\boldsymbol{V}^{(g)} \in \mathbb{R}^{n \times n}$ is positive definite and $\alpha, \beta \in \mathbb{R}^+$ are weighting factors (e.g., see [3, 10]). Let $\boldsymbol{v}^{(g)} \in \mathbb{R}$ be a step in the search space promising large fitness gain. To increase the probability that $\boldsymbol{v}^{(g)}$ is sampled in the next iteration, the rank-one update

$$\boldsymbol{C}^{(g+1)} = \alpha \boldsymbol{C}^{(g)} + \beta \boldsymbol{v}^{(g)} {\boldsymbol{v}^{(g)}}^T \tag{1}$$

can be used. This update rule shifts the mutation distribution towards the line distribution $\mathcal{N}\left(\boldsymbol{0}, \boldsymbol{v}^{(g)} {\boldsymbol{v}^{(g)}}^T\right)$, which is the distribution with the highest probability to generate $\boldsymbol{v}^{(g)}$ among all normal distributions with zero mean [3].

In general, each factorizing of a covariance matrix requires $O(n^3)$ operations. Thus, in an ES with additive covariance matrix update the Cholesky factorization of the covariance matrix is the computationally dominating factor apart from the fitness function evaluations. In [9] we therefore proposed not to factorize the covariance matrix, but to use an incremental rank-one update rule for the Cholesky factorization. This reduces the computational complexity to $O(n^2)$. The idea is not to compute the covariance matrix explicitly, but to operate on Cholesky factors only. Setting $\boldsymbol{v}^{(g)} = \boldsymbol{A}^{(g)} \boldsymbol{z}^{(g)}$ with $\boldsymbol{z}^{(g)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ we can rewrite the rank-one update of the covariance matrix equation (1) as

$$\boldsymbol{C}^{(g+1)} = \alpha \boldsymbol{C}^{(g)} + \beta \boldsymbol{A}^{(g)} \boldsymbol{z}^{(g)} \left[\boldsymbol{A}^{(g)} \boldsymbol{z}^{(g)}\right]^T \ . \tag{2}$$

Using the following theorem, we turn this update for $\boldsymbol{C}^{(g)}$ into an update for $\boldsymbol{A}^{(g)}$.

Theorem 1 ([9]). *Let $\boldsymbol{C}_t \in \mathbb{R}^{n \times n}$ be a symmetric nonnegative definite matrix with Cholesky factorization $\boldsymbol{C}_t = \boldsymbol{A}_t {\boldsymbol{A}_t}^T$. Assuming that $\boldsymbol{C}_t$ is updated using*

$$\boldsymbol{C}_{t+1} = \alpha \boldsymbol{C}_t + \beta \boldsymbol{v}_t \boldsymbol{v}_t^T,$$

with $\boldsymbol{v}_t = \boldsymbol{A}_t \boldsymbol{z}_t$*, where* $\boldsymbol{z}_t$ *is a column vector and* $\alpha, \beta \in \mathbb{R}^+$*. Then, the Cholesky factorization* $\boldsymbol{C}_{t+1} = \boldsymbol{A}_{t+1} {\boldsymbol{A}_{t+1}}^T$ *is given by*

$$\boldsymbol{A}_{t+1} = \sqrt{\alpha}\boldsymbol{A}_t + \frac{\sqrt{\alpha}}{\|\boldsymbol{z}_t\|^2} \left(\sqrt{1 + \frac{\beta}{\alpha}\|\boldsymbol{z}_t\|^2} - 1 \right) [\boldsymbol{A}_t \boldsymbol{z}_t] \, \boldsymbol{z}_t^T \ .$$

The new update rule guarantees a positive-definite covariance matrix. The numerical stability of the new update is likely to be better than an update requiring decompositions (e.g., see the discussion in [11, chapter 6]).

3 Generational and Steady-state Multi-objective Selection

Our multi-objective selection schemes is based on the non-dominated sorting approach used in NSGA-II [12, 6] and the selection scheme used in SMS-EMOA [7, 8].

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 $\mathrm{ndom}(A) = \{a \in A \,|\, \nexists a' \in A : a' \prec a\}$, where $a' \prec a$ means that a' dominates a. The Pareto front of A is then given by $\{(f_1(a), \ldots, f_M(a)) \,|\, a \in \mathrm{ndom}(A)\}$, where the f_i are the M real-valued objective functions. 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 (cf. [6, 9]). Formally, let $\mathrm{dom}_0(A) = A$, $\mathrm{dom}_l(A) = \mathrm{dom}_{l-1}(A) \setminus \mathrm{ndom}_l(A)$, and $\mathrm{ndom}_l(A) = \mathrm{ndom}(\mathrm{dom}_{l-1}(A))$ for $l \in \{1, \ldots\}$. For $a \in A$ we define the level of non-dominance $\mathrm{r}(a, A)$ to be i iff $a \in \mathrm{ndom}_i(A)$.

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 particular in real-valued optimization of continuous objective functions) after some generations there are more non-dominated solutions in the population than solutions to be selected. We consider the contributing hypervolume as second sorting criterion, which gave better results than the crowding-distance [6] in the experiments in [9]. The hypervolume measure or $\mathcal{S}$-metric was introduced by [13] in the domain of evolutionary MOO. It can be defined as the Lebesgue measure Λ (i.e., the volume) of the union of hypercuboids in the objective space:

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

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 rank $s(a, A')$ of an individual a can be defined recursively based on its contribution to the hypervolume, where ties are broken at random. The individual contributing least to the hypervolume of A' gets the worst rank. 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. We call $a \in A'$ a boundary element if $\Delta_{\mathcal{S}}(a, A')$ depends on the choice of the reference point a_{ref}. We choose a_{ref} such that all elements in A' dominate a_{ref} and that for any boundary element $a \in A'$ and any non boundary element $a' \in A'$ we have $\Delta_{\mathcal{S}}(a, A') > \Delta_{\mathcal{S}}(a', A')$. That is, the individuals at the "boundaries" of the Pareto front of A' are preferably selected. Let a lower rank be worse. Formally (assuming that argmin breaks ties randomly), for $a \in \text{ndom}(A')$ we have $s(a, A') = 1$ if $a = \text{argmin}_{a' \in A'}\{\Delta_{\mathcal{S}}(a', A')\}$ and $s(a, A') = k$ if $a = \text{argmin}_{a' \in A'}\{\Delta_{\mathcal{S}}(a', A' \setminus \{a'' \mid s(a'', A') < k\})\}$. Based on this ranking and the level of non-dominance we define the relation

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

for $a, a' \in A$. That is, a is better than a' when compared using $\prec_A$ 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.

In the following, we consider three reproduction and selection schemes based on this ranking. First, in *generational* selection $(\mu+\mu)$ as described in [9] each of the μ parents generates one offspring per generation. The resulting 2μ individuals are sorted as described above and the μ best form the next parent population.

Then we consider two *steady-state* [14, 15] selection schemes, in which only a single parent creates one offspring per generation. If this offspring a is better than the worst individual in the parent population A w.r.t. $\prec_{A \cup \{a\}}$, a replaces the worst individual. Otherwise the offspring is discarded. The two steady state variants differ in the way the parent of the offspring is selected. In the first variant $(\mu+1)$, the parent is chosen uniformly at random from A. In the second version $(\mu_{\prec}+1)$, the parent is chosen uniformly at random from $\text{ndom}(A)$. The idea behind the second approach, which can be regarded as more greedy, is that it is more promising to allow reproduction of non-dominated individuals than of dominated. Because the individuals $A \setminus \text{ndom}(A)$ do not influence the evolutionary process anymore, they can be discarded. That is, the second variant is an algorithm with varying population (or archive) size. Only non-dominated individuals remain in the population while the size of the population is still upper bounded by μ.

4 MO-CMA-ES

In the MO-CMA-ES with standard covariance matrix update the kth individual in generation g is a 5-tuple denoted by $a_k^{(g)} = [\boldsymbol{x}_k^{(g)}, \overline{p}_{\text{succ},k}^{(g)}, \sigma_k^{(g)}, \boldsymbol{p}_{c,k}^{(g)}, \boldsymbol{C}_k^{(g)}]$. Here, $\boldsymbol{x}_k^{(g)} \in \mathbb{R}^n$ is the point in the search space, $\overline{p}_{\text{succ},k}^{(g)} \in \mathbb{R}_0^+$ the average success rate, $\sigma_k^{(g)} \in \mathbb{R}^+$ the global step size, $\boldsymbol{p}_{c,k}^{(g)} \in \mathbb{R}^n$ the evolution path, and $\boldsymbol{C}_k^{(g)} \in \mathbb{R}^{n \times n}$ the covariance matrix.

The standard version of the generational MO-CMA-ES reads as follows (ignoring lines 5b and 10b for a moment):

Algorithm 1: generational MO-CMA

1 $g = 0$, initialize $a_k^{(g)}$ for $k = 1, \ldots, \mu$
2 **repeat**
3 **for** $k = 1, \ldots, \mu$ **do**
4 $a'^{(g+1)}_k \leftarrow a_k^{(g)}$
5a $\boldsymbol{x}'^{(g+1)}_k \sim \mathcal{N}\left(\boldsymbol{x}_k^{(g)}, {\sigma_k^{(g)}}^2 \boldsymbol{C}_k^{(g)}\right)$
5b $\boldsymbol{x}'^{(g+1)}_k \sim \mathcal{N}\left(\boldsymbol{x}_k^{(g)}, {\sigma_k^{(g)}}^2 \boldsymbol{A}_k^{(g)} {\boldsymbol{A}_k^{(g)}}^T\right)$
6 $Q^{(g)} = \left\{ a'^{(g+1)}_k, a_k^{(g)} \,\middle|\, 1 \leq k \leq \mu \right\}$
7 **for** $k = 1, \ldots, \mu$ **do**
8 $\texttt{updateStepSize}\left(a_k^{(g)}, \mathbb{1}[a'^{(g+1)}_k \prec_{Q^{(g)}} a_k^{(g)}]\right)$
9 $\texttt{updateStepSize}\left(a'^{(g+1)}_k, \mathbb{1}[a'^{(g+1)}_k \prec_{Q^{(g)}} a_k^{(g)}]\right)$
10a $\texttt{updateCovariance}\left(a'^{(g+1)}_k, \dfrac{\boldsymbol{x}'^{(g+1)}_k - \boldsymbol{x}_k^{(g)}}{\sigma_k^{(g)}}\right)$
10b $\texttt{updateCholesky}\left(a'^{(g+1)}_k, \dfrac{\boldsymbol{x}'^{(g+1)}_k - \boldsymbol{x}_k^{(g)}}{\sigma_k^{(g)}}\right)$
11 **for** $i = 1, \ldots, \mu$ **do** $a_i^{(g+1)} \leftarrow Q^{(g)}_{\prec:i}$
12 $g \leftarrow g + 1$
until stopping criterion is met

Each of the μ parents generates one offspring (lines 3–5) . Parents and offspring form the set $Q^{(g)}$ (line 6). The step sizes of a parent and its offspring are updated depending on whether the mutations were successful (lines 7–9), that is, whether the offspring is better than the parent according to the relation $\prec_{Q^{(g)}}$ (the indicator function $\mathbb{1}[\cdot]$ is 1 if its argument is true and 0 otherwise).

The covariance matrix of the offspring (line 10a) is adjusted 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+1)-CMA-ES, see [5, 9] for details. The best μ individuals in $Q^{(g)}$ sorted by $\prec_{Q^{(g)}}$ form the next parent generation (line 11, where $Q^{(g)}_{\prec:i}$ is the ith best offspring in $Q^{(g)}$ w.r.t. $\prec_{Q^{(g)}}$).

The update rule for the global step size is rooted in the 1/5-success-rule proposed in [1] and is an extension from the rule proposed in [4]:

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

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

2 $\sigma \leftarrow \sigma \cdot \exp\left(\frac{1}{d}\frac{\overline{p}_{\text{succ}} - p_{\text{succ}}^{\text{target}}}{1 - p_{\text{succ}}^{\text{target}}}\right)$

This rule implements the well-known heuristic that the step size should be increased if the success rate of mutation is high, and the step size should be decreased if the success rate is low. The damping parameter d controls the rate of the step size adaptation.

Then the covariance matrices are adapted (see main routine line 10a):

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 $\quad \boldsymbol{p}_c \leftarrow (1 - c_c)\boldsymbol{p}_c + \sqrt{c_{\text{c}}(2 - c_{\text{c}})}\,\boldsymbol{x}_{\text{step}}$

3 $\quad \boldsymbol{C} \leftarrow (1 - c_{\text{cov}})\boldsymbol{C} + c_{\text{cov}} \cdot \boldsymbol{p}_c \boldsymbol{p}_c^{\,T}$

4 **else**

5 $\quad \boldsymbol{p}_c \leftarrow (1 - c_c)\boldsymbol{p}_c$

6 $\quad \boldsymbol{C} \leftarrow (1 - c_{\text{cov}})\boldsymbol{C} + c_{\text{cov}} \cdot \left(\boldsymbol{p}_c \boldsymbol{p}_c^{\,T} + c_{\text{c}}(2 - c_{\text{c}})\boldsymbol{C}\right)$

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. If the smoothed success rate $\overline{p}_{\text{succ}}$ is low, the update of $\boldsymbol{p}_c$ is accomplished with 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 $\sqrt{c_{\text{c}}(2 - c_{\text{c}})}$ normalizes the variance of $\boldsymbol{p}_c$ viewed as a random variable (see [3]). The evolution path $\boldsymbol{p}_c$ is then used to update the covariance matrix. The new covariance matrix is a weighted mean of the old 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 $c_{\text{c}}(2 - c_{\text{c}})\boldsymbol{C}$ (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 [9] and used in this paper, are: $d = 1 + n/2$, $p_{\text{succ}}^{\text{target}} = (5 + \sqrt{1/2})^{-1}$, $c_p = p_{\text{succ}}^{\text{target}}/(2 + p_{\text{succ}}^{\text{target}})$, $c_{\text{c}} = 2/(n + 2)$, $c_{\text{cov}} = 2/(n^2 + 6)$, and $p_{\text{thresh}} = 0.44$.

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 $\boldsymbol{C} = \boldsymbol{I}$. The initial candidate solution $\boldsymbol{x} \in \mathbb{R}^n$ and the initial $\sigma \in \mathbb{R}^+$ must be chosen problem dependent. The optimum should presumably be within the cube $\boldsymbol{x} \pm 2\,\sigma\,(1, \ldots, 1)^T$.

4.1 Cholesky Update

In the standard generational MO-CMA-ES, denoted by $(\mu+\mu)$ in the following, there are up to μ covariance updates (in an efficient implementation only covariance matrices of those offspring that will be in the next parent population are updated). Therefore, computation time could be significantly reduced using the concepts described in [9] and Section 2—if objective function evaluation is fast and the dimensionality n of the search space is large.

In the generational MO-CMA with "Cholesky update", denoted by $(\mu+\mu)_{\text{chol}}$ in the following, the kth individual in generation g consists of a 4-tuple $a_k^{(g)} = [\boldsymbol{x}_k^{(g)}, \overline{p}_{\text{succ},k}^{(g)}, \sigma_k^{(g)}, \boldsymbol{A}_k^{(g)}]$, where the Cholesky factor $\boldsymbol{A}_k^{(g)} \in \mathbb{R}^{n\times n}$ is stored instead of the covariance matrix. The update of the Cholesky factor is given by applying Theorem 1:

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

1 **if** $\overline{p}_{\text{succ}} < p_{\text{thresh}}$ **then**

2 $$\boldsymbol{A} \leftarrow \sqrt{1-c_{\text{cov}}}\,\boldsymbol{A} + \frac{\sqrt{1-c_{\text{cov}}}}{\|\boldsymbol{x}_{\text{step}}\|^2}\left(\sqrt{1+\frac{c_{\text{cov}}\|\boldsymbol{x}_{\text{step}}\|^2}{1-c_{\text{cov}}}}-1\right)\boldsymbol{A}\boldsymbol{x}_{\text{step}}\boldsymbol{x}_{\text{step}}^T$$

Because this update rule does not work with an evolution path (see [5]), the covariance adaptation usually slows down in terms of the number of generations needed to learn the metric of the underlying problem [9]. However, how strong this effect is depends on the optimization problem.

The algorithmic description of the $(\mu+\mu)$ is obtained from Algorithm 1 using lines 5b and 10b instead of 5a and 10a. The replacement allows for a simple implementation of the $(\mu+\mu)$, because it avoids the otherwise necessary matrix decomposition of the standard generational MO-CMA-ES.

4.2 Steady-state Selection

In steady-state selection schemes only one offspring is generated per generation. Here, we consider two different variants of steady-state selection. The first one, denoted by $(\mu_{\prec}+1)$ in the remainder of this article, selects the parent among the non-dominated solutions in the population. As the dominated solutions in the parent population do not influence the evolutionary dynamics, they can be removed from the population. Thus, this variant can be viewed as an evolutionary algorithm with adaptive population size, where the number of individuals equals the number of non-dominated solutions upper bounded by μ.

The second steady-state algorithm, denoted by $(\mu+1)$ in the following, considers all μ members of the population as potential parents and hence is less greedy than the first variant. This corresponds to the selection scheme used in [7, 8].

Both variants are described in Algorithm 2 and are quite similar to the generational MO-CMA. The main difference is the selection of the parent for reproduction in line 4a for the $(\mu_{\prec}+1)$ and in line 4b for the $(\mu+1)$, respectively.

Algorithm 2: steady-state MO-CMA

$g = 0$, initialize $a_k^{(g)}$ for $k = 1, \ldots, \mu$
repeat
  $Q^{(g)} = \left\{ a_k^{(g)} \,\middle|\, 1 \leq k \leq \mu \right\}$
  $i \leftarrow \mathcal{U}(1, |\operatorname{ndom}(Q^{(g)})|)$
  $i \leftarrow \mathcal{U}(1, |Q^{(g)}|)$
  $a^{(g+1)} \leftarrow Q^{(g)}_{\prec:i}$
  $a'^{(g+1)} \leftarrow a^{(g+1)}$
  $\boldsymbol{x}'^{(g+1)} \sim \mathcal{N}\left(\boldsymbol{x}^{(g+1)}, {\sigma^{(g)}}^2 \boldsymbol{C}^{(g)}\right)$
  $Q^{(g)} \leftarrow Q^{(g)} \cup \{a'^{(g+1)}\}$
  `updateStepSize` $\left(a^{(g)}, \mathbb{1}[a'^{(g)} \prec_{Q^{(g)}} a^{(g)}]\right)$
  `updateStepSize` $\left(a'^{(g+1)}, \mathbb{1}[a'^{(g)} \prec_{Q^{(g)}} a^{(g)}]\right)$
  `updateCovariance` $\left(a'^{(g+1)}, \dfrac{\boldsymbol{x}'^{(g+1)} - \boldsymbol{x}^{(g)}}{\sigma^{(g)}}\right)$
  for $i = 1, \ldots, \mu$ **do** $a_i^{(g+1)} \leftarrow Q^{(g)}_{\prec:i}$
  $g \leftarrow g + 1$
until stopping criterion is met

5 Experiments

In the following, we empirically evaluate the different variants of the MO-CMA-ES presented in the previous section. In [9] we compared the generational MO-CMA-ES with other multi-objective evolutionary algorithms, namely NSGA-II and the multi-criteria differential evolution algorithm NSDE [16]. Because of the good performance of the MO-CMA-ES in [9], we do not consider any other reference algorithm in the present study.

5.1 Evaluating the Performance of MOO Algorithms

Many ways of measuring the performance of MOO algorithms have been proposed. Here we follow recommendations in [17] and use unary quality indicators, for a detailed description of the methods we refer to [18, 17, 5].

An unary quality indicator assigns a real valued quality to a set of solutions. Here, the hypervolume indicator [13] and the ϵ-indicator [18] are measured. We use the performance assessment tools contributed to the PISA [19] software package with standard parameters. The hypervolume indicator w.r.t. reference set A_{ref} (see below) is defined as $\mathcal{I}_{\mathcal{S},A_{\text{ref}}}(A) = \mathcal{S}_{a_{\text{ref}}}(A_{\text{ref}}) - \mathcal{S}_{a_{\text{ref}}}(A)$ where a_{ref} denotes a (hypothetical) reference point having in each objective an objective function value worse than all considered individuals. The additive unary ϵ-indicator $\mathcal{I}_{\epsilon,A_{\text{ref}}}$

w.r.t. reference set A_{ref} is defined as the smallest offset by which the fitness values of the elements in A have to be shifted such that the resulting set dominates A_{ref}. Both a small $\mathcal{I}_{\mathcal{S},A_{\text{ref}}}$ and a small $\mathcal{I}_{\epsilon,A_{\text{ref}}}$ are preferable.

Before the performance indicators are computed, the data are normalized. We want to compare k algorithms on a particular optimization problem after g_1 and g_2 fitness evaluations (here, 25000 and 50000) and we assume that we have conducted t trials. We consider the non-dominated individuals of the union of all $2 \cdot k \cdot t$ populations after g_1 and g_2 evaluations. These individuals make up the reference set A_{ref}. 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. 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. Note that the set A_{ref} is comprised of rather well performing individuals, whereas the point a_{ref} has bad objective function values.

5.2 Benchmark Functions

Table 1. 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
ELLI_1	10	$[-10,10]$	$f_1(\boldsymbol{y}) = \frac{1}{a^2 n}\sum_{i=1}^{n} a^{2\frac{i-1}{n-1}} y_i^2$ $f_2(\boldsymbol{y}) = \frac{1}{a^2 n}\sum_{i=1}^{n} a^{2\frac{i-1}{n-1}} (y_i - 2)^2$
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	$[-10,10]$	$f_1(\boldsymbol{y}) = \frac{1}{a^2 n}\left[y_1^2 + \sum_{i=2}^{n-1} a y_i^2 + a^2 y_n^2\right]$ $f_2(\boldsymbol{y}) = \frac{1}{a^2 n}\left[(y_1 - 2)^2 + \sum_{i=2}^{n-1} a\,(y_i - 2)^2 + a^2 (y_n - 2)^2\right]$
CIGTAB_2	10	$[-10,10]$	$f_1(\boldsymbol{y}) = \frac{1}{a^2 n}\left[y_1^2 + \sum_{i=2}^{n-1} a y_i^2 + a^2 y_n^2\right]$ $f_2(\boldsymbol{z}) = \frac{1}{a^2 n}\left[(z_1 - 2)^2 + \sum_{i=2}^{n-1} a\,(z_i - 2)^2 + a^2 (z_n - 2)^2\right]$

We consider three groups of test functions. The first group comprises six common benchmark problems taken from the literature, namely the function FON proposed in [20] and the test functions ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 proposed in [21]. All functions have box constraints also given in the table. As most components of the optimal solution lie on the boundary of these box constraints, we question the general relevance of these test functions. In accordance with [22, 23], we believe that "rotated" functions, which are less aligned with the coordinate system of the search space, are more appropriate.

This led to the definition of the two other groups of benchmark functions, see [5] for details.

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 1. 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 eigenvalue spectrum of $\boldsymbol{Q}$. In each optimization run the coordinate system of the objective functions is changed by a random choice of $\boldsymbol{O}_m$ (see [9] for details). In the case of the test functions ELLI_1 and CIGTAB_1 the same rotation 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.

The third group of problems shown in Table 2 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 coordinate. 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. 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. The function $h : \mathbb{R} \to [0, 1]$ is monotonic and emulates the original variable boundary $x_1 \in [0, 1]$. Similar, the function $h_g : \mathbb{R} \to \mathbb{R}_0^+$ emulates the original lower variable boundary of $x_i \geq 0$ for $i = 2, \ldots, n$.

5.3 Experiments

We used the same parameters in the MO-CMA-ES as in [9]. For the functions FON, ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 we set $\sigma^{(0)}$ equal to 60 % of the feasible region $x_2^u - x_2^l$ (we rescaled the first component of ZDT4 to $[-5, 5]$). In the unconstrained problems, Table 1, we set $\sigma^{(0)}$ equal to 60 % of the initialization range of one component. In all algorithms the population size μ was set to 100. For each test problem, 100 trials were conducted per algorithm.

6 Results and Discussion

The results are summarized in Tables 3 to 5. In general, the MO-CMA with Cholesky update performed worse than the MO-CMA using an evolution path, although the differences are often not significant. After 50000 evaluations the methods differ significantly in at least one indicator on FON, ELLI_2, CIGTAB_1,

Table 2. 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 $o_{11} = 1$. The auxiliary functions are defined as $h : \mathbb{R} \to [0,1], x \mapsto \left(1 + \exp\left(\frac{-x}{\sqrt{n}}\right)\right)^{-1}$, $h_f : \mathbb{R} \to \mathbb{R}, x \mapsto \begin{cases} x & \text{if } |y_1| \leq y_{\max} \\ 1 + |y_1| & \text{otherwise} \end{cases}$, and $h_g : \mathbb{R} \to \mathbb{R}_0^+, x \mapsto \frac{x^2}{|x|+0.1}$.

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

CIGTAB_2, IHR1, and IHR6. On the multi-modal problems, where the results are dominated by the global search performance, the $(\mu+\mu)_{\text{chol}}$ results look slightly better than those with $(\mu+\mu)$, while the differences are not apparent in our statistics. After 25000 evaluations, where the covariance matrix adaptation did not pay off yet, the differences are not significant, except for the FON function, where covariance matrix adaptation is faster due to the low dimensionality. These results are in accordance with those in [9].

The newly developed covariance matrix update rule reduces the computational complexity of the rank-one covariance matrix adaptation from $O(n^3)$ to $O(n^2)$. This is a significant improvement on high dimensional, but fast computable fitness functions. However, in practice it is not necessary to perform the covariance matrix decomposition, as required in the original covariance matrix

Table 3. Median results over 100 trials on standard benchmark functions after 25000 and 50000 evaluations, respectively. Superscripts indicate significant differences; I: $(\mu+\mu)$, ..., IV: $(\mu+1)$, two-sided Wilcoxon rank sum test, normal font $p < 0.001$, slanted $p < 0.01$.

hypervolume indicator						
algorithm	FON	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
25000 evaluations						
$(\mu+\mu)$	0.00480^{II}	0.00377	0.00483	0.00139	0.17444^{III}	0.00052
$(\mu+\mu)_{\text{chol}}$	0.00482	0.00377	0.00484	0.00140	$\underline{0.16218}^{III}$	0.00052
$(\mu_{\prec}+1)$	$\underline{0.00448}^{I,II,IV}$	$\underline{0.00357}^{I,II,IV}$	$\underline{0.00451}^{I,II,IV}$	$\underline{0.00129}^{I,II,IV}$	0.39748	$\underline{0.00050}^{I,II}$
$(\mu+1)$	$0.00448^{I,II}$	$0.00363^{I,II}$	$0.00466^{I,II}$	0.00137	0.17113^{III}	$0.00050^{I,II}$
50000 evaluations						
$(\mu+\mu)$	0.00473^{II}	0.00365	0.00472	0.00132	$\underline{0.12979}^{III}$	0.00052
$(\mu+\mu)_{\text{chol}}$	0.00476	0.00365	0.00473	0.00134	0.13068^{III}	0.00052
$(\mu_{\prec}+1)$	$\underline{0.00448}^{I,II}$	$\underline{0.00349}^{I,II}$	$\underline{0.00445}^{I,II,IV}$	$\underline{0.00125}^{I,II,IV}$	0.35486	$\underline{0.00050}^{I,II}$
$(\mu+1)$	$0.00448^{I,II}$	$0.00350^{I,II}$	$0.00452^{I,II}$	$0.00129^{I,II}$	0.15571^{III}	$0.00050^{I,II}$
ϵ-indicator						
algorithm	FON	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
25000 evaluations						
$(\mu+\mu)$	0.00698	0.00624	0.00707	0.00345	0.16344^{III}	0.00147
$(\mu+\mu)_{\text{chol}}$	0.00695	0.00615	0.00703	0.00342	$\underline{0.15132}^{III}$	0.00149
$(\mu_{\prec}+1)$	$\underline{0.00497}^{I,II}$	$\underline{0.00491}^{I,II}$	$\underline{0.00541}^{I,II,IV}$	$\underline{0.00285}^{I,II,IV}$	0.35884	$\underline{0.00106}^{I,II}$
$(\mu+1)$	$0.00501^{I,II}$	$0.00501^{I,II}$	$0.00569^{I,II}$	$0.00306^{I,II}$	0.15800^{III}	$0.00106^{I,II}$
50000 evaluations						
$(\mu+\mu)$	0.00694	0.00607	0.00699	0.00354	0.13596^{III}	0.00149
$(\mu+\mu)_{\text{chol}}$	0.00690	0.00613	0.00697	0.00355	$\underline{0.12269}^{III}$	0.00150
$(\mu_{\prec}+1)$	$0.00487^{I,II}$	$0.00460^{I,II}$	$\underline{0.00514}^{I,II,IV}$	$\underline{0.00269}^{I,II}$	0.33336	$\underline{0.00101}^{I,II}$
$(\mu+1)$	$\underline{0.00487}^{I,II}$	$\underline{0.00460}^{I,II}$	$0.00526^{I,II}$	$0.00273^{I,II}$	0.14155^{III}	$0.00101^{I,II}$

adaptation, each generation, but only every τ generations. Then the computational complexity becomes $O(n^3/\tau + n^2)$. For $\tau = o(n)$ the Cholesky approach is still faster for large n, while $\tau = \omega(n)$ is not advisable. Apart from that, the new update rule is much simpler to implement (e.g., allowing for easy implementations in hardware and in low level programming languages) and is completely specified without any hidden, numerically involved procedures such as a singular value decomposition.

On the unimodal problems, the steady-state algorithms perform better than the generational MO-CMA-ES. Here the greedy steady-state $(\mu_{\prec}+1)$-MO-CMA-ES performs best. But on the multi-modal problems, the generational algorithms are superior. However, the $(\mu+1)$-MO-CMA-ES is not significantly worse, whereas the performance of the greedy $(\mu_{\prec}+1)$-MO-CMA-ES is so bad that it should not be considered as an alternative to the generational MO-CMA despite its good performance on the other test problems.

Thus, we recommend the $(\mu+1)$-MO-CMA-ES. The selection strategy of this variant is equal to the strategy in the SMS-EMOA proposed in [7, 8], only the

Table 4. Median results over 100 trials on rotated quadratic benchmark functions after 25000 evaluations and 50000 evaluations, respectively.

hypervolume indicator				
algorithm	ELLI_1	ELLI_2	CIGTAB_1	CIGTAB_2
25000 evaluations				
$(\mu+\mu)$	0.003931	0.000037	0.003466	0.000037
$(\mu+\mu)_{\text{chol}}$	0.004050	0.000037	0.003486	0.000042
$(\mu_{\prec}+1)$	$\underline{0.003771}^{II}$	$\underline{0.000018}^{\text{I},\text{II},\text{IV}}$	$\underline{0.003092}^{\text{I},\text{II},\text{IV}}$	$\underline{0.000015}^{\text{I},\text{II},\text{IV}}$
$(\mu+1)$	0.003963	0.000039	$0.003132^{\text{I},\text{II}}$	$0.000031^{\text{I},\text{II}}$
50000 evaluations				
$(\mu+\mu)$	$\underline{0.003468}^{IV}$	0.000012^{II}	0.003382^{II}	0.000004
$(\mu+\mu)_{\text{chol}}$	0.003611	0.000019	0.003400	0.000004
$(\mu_{\prec}+1)$	0.003575	$\underline{0.000005}^{\text{I},\text{II},\text{IV}}$	$\underline{0.003068}^{\text{I},\text{II}}$	$\underline{0.000002}^{\text{I},\text{II},\text{IV}}$
$(\mu+1)$	0.003592	0.000012^{II}	$0.003077^{\text{I},\text{II}}$	0.000004
ϵ-indicator				
algorithm	ELLI_1	ELLI_2	CIGTAB_1	CIGTAB_2
25000 evaluations				
$(\mu+\mu)$	0.006015	0.000120	0.005835	0.000196
$(\mu+\mu)_{\text{chol}}$	0.005981	0.000134	0.005897	0.000214
$(\mu_{\prec}+1)$	$\underline{0.004717}^{\text{I},\text{II},\text{IV}}$	$\underline{0.000060}^{\text{I},\text{II},\text{IV}}$	$\underline{0.004294}^{\text{I},\text{II},\text{IV}}$	$\underline{0.000145}^{\text{I},\text{II},\text{IV}}$
$(\mu+1)$	$0.005191^{\text{I},\text{II}}$	0.000122	$0.004463^{\text{I},\text{II}}$	$0.000185^{\text{I},\text{II}}$
50000 evaluations				
$(\mu+\mu)$	0.005742	0.000056^{II}	0.005779	0.000149^{II}
$(\mu+\mu)_{\text{chol}}$	0.005823	0.000073	0.005759	0.000152
$(\mu_{\prec}+1)$	$\underline{0.004261}^{\text{I},\text{II}}$	$\underline{0.000045}^{\text{I},\text{II},\text{IV}}$	$\underline{0.004030}^{\text{I},\text{II}}$	$\underline{0.000142}^{\text{I},\text{II},\text{IV}}$
$(\mu+1)$	$0.004313^{\text{I},\text{II}}$	0.000047^{II}	$0.004030^{\text{I},\text{II}}$	0.000148^{II}

variation operators and the strategy adaptation differ between SMS-EMOA and $(\mu+1)$-MO-CMA.

References

1. Rechenberg, I.: Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution. Werkstatt Bionik und Evolutionstechnik. Frommann-Holzboog, Stuttgart (1973)
2. Schwefel, H.P.: Evolution and Optimum Seeking. Sixth-Generation Computer Technology Series. John Wiley & Sons (1995)
3. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9**(2) (2001) 159–195
4. Kern, S., Müller, S., Hansen, N., Büche, D., Ocenasek, J., Koumoutsakos, P.: Learning probability distributions in continuous evolutionary algorithms – A comparative review. Natural Computing **3** (2004) 77–112
5. Igel, C., Hansen, N., Roth, S.: Covariance matrix adaptation for multi-objective optimization. Evolutionary Computation (2006) Accepted.

Table 5. Median results over 100 trials on new rotated benchmark functions after 25000 evaluations and 50000 evaluations, respectively.

hypervolume indicator						
algorithm	ZDT4'	IHR1	IHR2	IHR3	IHR4	IHR6
25000 evaluations						
$(\mu+\mu)$	$\underline{0.18487}^{\text{III}}$	0.00750	0.04023	0.02678	$\underline{0.00484}^{\text{III}}$	0.17635
$(\mu+\mu)_{\text{chol}}$	0.19488^{III}	0.00759	0.03960	0.02686	0.00496^{III}	0.18078
$(\mu_{\prec}+1)$	0.48206	$\underline{0.00119}^{\text{I,II,IV}}$	$\underline{0.03877}^{\text{I,II,IV}}$	$\underline{0.02634}^{\text{IV}}$	0.01753	$\underline{0.03198}^{\text{I,II,IV}}$
$(\mu+1)$	0.21022^{III}	0.00813	$0.03927^{\text{I,II}}$	0.02654	0.00521^{III}	$0.13856^{\text{I,II}}$
50000 evaluations						
$(\mu+\mu)$	$\underline{0.14438}^{\text{III,IV}}$	0.00161^{II}	0.03799	0.02633	$0.00415^{\text{III,IV}}$	0.03522^{II}
$(\mu+\mu)_{\text{chol}}$	0.16716^{III}	0.00658	0.03789^{I}	0.02633	$\underline{0.00402}^{\text{III,IV}}$	0.03928
$(\mu_{\prec}+1)$	0.42775	$\underline{0.00082}^{\text{I,II,IV}}$	$\underline{0.03785}^{\text{I,II}}$	$\underline{0.02633}^{\text{IV}}$	0.01746	$\underline{0.02749}^{\text{I,II}}$
$(\mu+1)$	0.18228^{III}	$0.00115^{\text{I,II}}$	$0.03787^{\text{I,II}}$	0.02633	0.00501^{III}	$0.03034^{\text{I,II}}$
ϵ-indicator						
algorithm	ZDT4'	IHR1	IHR2	IHR3	IHR4	IHR6
25000 evaluations						
$(\mu+\mu)$	$\underline{0.18533}^{\text{III}}$	0.01440	0.14304	0.04360	$\underline{0.00526}^{\text{III}}$	0.18192
$(\mu+\mu)_{\text{chol}}$	0.19352^{III}	0.01446	0.14269	0.04367	0.00533^{III}	0.18782
$(\mu_{\prec}+1)$	0.45294	$\underline{0.00444}^{\text{I,II,IV}}$	$\underline{0.14268}^{\text{I,II,IV}}$	$\underline{0.04321}^{\text{IV}}$	0.01666	$\underline{0.06542}^{\text{I,II,IV}}$
$(\mu+1)$	0.20867^{III}	0.01515	$0.14284^{\text{I,II}}$	0.04340	0.00537^{III}	$0.15154^{\text{I,II}}$
50000 evaluations						
$(\mu+\mu)$	$\underline{0.14594}^{\text{III,IV}}$	0.00553^{II}	0.14066	0.04320	$\underline{0.00393}^{\text{III,IV}}$	0.05598^{II}
$(\mu+\mu)_{\text{chol}}$	0.16683^{III}	0.01345	0.14046	0.04320	0.00408^{III}	0.06493
$(\mu_{\prec}+1)$	0.37796	$\underline{0.00362}^{\text{I,II,IV}}$	$\underline{0.14043}^{\text{I,II}}$	$\underline{0.04320}$	0.01665	0.06490
$(\mu+1)$	0.17945^{III}	$0.00446^{\text{I,II}}$	0.14046	0.04320	0.00474^{III}	$\underline{0.05340}^{\text{II}}$

6. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2) (2002) 182–197
7. Emmerich, M., Beume, N., Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. In Coello Coello, C.A., Zitzler, E., Hernandez Aguirre, A., eds.: Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Volume 3410 of LNCS., Springer-Verlag (2005) 62–76
8. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research (2007) In press.
9. Igel, C., Suttorp, T., Hansen, N.: A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006), ACM Press (2006) 453–460
10. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation **11**(1) (2003) 1–18
11. Grewal, M.S., Andrews, A.P.: Kalman Filtering: Theory and Practice. Prentice-Hall (1993)

12. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley (2001)
13. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms – a comparative case study. In Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P., eds.: Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V), Springer-Verlag (1998) 292–301
14. Whitley, L.D.: The GENITOR algorithm and selection pressure: Why rank–based allocation of reproductive trials is best. In Schaffer, J.D., ed.: Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89), Morgan Kaufmann Publishers (1989) 116–121
15. Syswerda, G.: Uniform crossover in genetic algorithms. In Schaffer, J.D., ed.: Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89), Morgan Kaufmann Publishers (1989) 2–9
16. Iorio, A., Li, X.: Solving rotated multi-objective optimization problems using differential evolution. In Webb, G.I., Yu, X., eds.: Proceeding of the 17th Joint Australian Conference on Artificial Intelligence. Volume 3339 of LNCS., Spriner-Verlag (2005) 861–872
17. Knowles, J., Thiele, L., Zitzler, E.: A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK-Report 214, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich (2005)
18. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Grunert da Fonseca, V.: Performance assesment of multiobjective optimizers: An analysis and review. IEEE Transactions on Evolutionary Computation **7**(2) (2003) 117–132
19. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA – A platform and programming language independent interface for search algorithms. In Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L., eds.: Evolutionary Multi-Criterion Optimization (EMO 2003). Volume 2632 of LNCS., Springer-Verlag (2003) 494 – 508
20. Fonseca, C.M., Fleming, P.J.: 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) (1998) 38–47
21. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation **8**(2) (2000) 173–195
22. Deb, K., Sinha, A., Kukkonen, S.: Multi-objective test problems, linkages, and evolutionary methodologies. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006), ACM Press (2006) 1141–1148
23. Iorio, A.W., Li, X.: Rotated test problems for assessing the performance of multi-objective optimization algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006), ACM Press (2006) 683–690