Self-Adaptation for Multi-objective Evolutionary Algorithms

Dirk Büche, Sibylle Müller, and Petros Koumoutsakos

Institute of Computational Science, Swiss Federal Institute of Technology (ETH), CH - 8092 Zürich, Switzerland
E-mail: {*bueche, muellers, petros*}*@inf.ethz.ch*
http://www.icos.ethz.ch

Abstract. Evolutionary Algorithms are a standard tool for multi-objective optimization that are able to approximate the Pareto front in a single optimization run. However, for some selection operators, the algorithm stagnates at a certain distance from the Pareto front without convergence for further iterations.
We analyze this observation for different multi-objective selection operators. We derive a simple analytical estimate of the stagnation distance for several selection operators, that use the dominance criterion for the fitness assignment. Two of the examined operators are shown to converge with arbitrary precision to the Pareto front. We exploit this property and propose a novel algorithm to increase their convergence speed by introducing suitable self-adaptive mutation. This adaptive mutation takes into account the distance to the Pareto front. All algorithms are analyzed on a 2- and 3-objective test function.

1 Introduction

Real-world optimization problems often include multiple and conflicting objectives. A solution to such a problem is often a compromise between the objectives, since no solution can be found that is ideal to all objectives. The set of the best compromise solutions is referred to as the Pareto-ideal set, characterized by the fact that starting from a solution within the set, one objective can only be improved at the expense of at least one other objective.
Evolutionary Algorithms (EAs) are a standard tool for Pareto optimization, since their population-based search allows approximating the Pareto front in a single optimization run. EAs operate by evolving the population in a cooperative search towards the Pareto front. They incorporate biologically inspired operators such as mutation, recombination, and fitness-based selection.
In Pareto optimization, recent research has focused on multi-objective selection operators and in particular fitness assignment techniques. Various selection operators are compared in the literature [1–3] and according to Van Veldhuizen and Lamont [4], the dominance criterion in combination with niching techniques is one of the most efficient techniques for the fitness assignment. This group of algorithms is referred to by Horn [5] as "Cooperative Population Searches

with Dominance Criterion" and two prominent representatives are SPEA [1] and NSGA-II [6].
While these algorithms perform well in a number of test problems, we observe a stagnation in the convergence of these algorithms at a certain distance from the Pareto front. The distance is dependent on the selection operator as usually these algorithms do not employ any mutation or recombination operators. In this paper we estimate this stagnation distance by deriving an analytical solution for a simplified Pareto front. This raises the question, which selection operators are able to converge to the Pareto front and in addition, which operators converge efficiently?
Two alternatives to the dominance criterion are the Constraint Method-based Evolutionary Algorithm (CMEA) [2] and Subdivision Method (SDM) [7]. These algorithms perform selection by optimizing one objective, while the other objectives are treated as constraints. They are able to converge to the Pareto front for certain test cases.
In conjunction with the selection operator, the mutation and recombination operators are important for an efficient convergence and should adapt while converging towards the Pareto front. In recent years, some efforts have been made in order to apply adaptation in multi-objective optimization. Kursawe [8] and Laumanns *et al.* [9] developed two implementations of self-adaptation [10]. Kursawe performs selection based on a randomly chosen objective. In his work each individual contains a separate vector of design variables and step sizes for each objective (polyploid individuals). Laumanns *et al.* assign a single step size to each individual, which yields an isotropic mutation distributions.
Sbalzarini et al. [11] use a simple self-adaptation scheme for mutating step sizes. Each individual in the population is assigned an individual step size for each design variable. Abbass [12] implemented a Pareto optimization algorithm with recombination and mutation based on the Differential Evolution of Storn and Price [13]. He used self-adaptation in order to find appropriate crossover and mutation rates (probabilities). Büche *et al.* [14] trained self-organizing maps (SOMs) [15] on the currently nondominated solutions, and recombination was performed within the network. The mutation strength was related to the distance between the neurons in the SOM.
Compared to single objective optimization, applying self-adaptive mutation to multi-objective selection operators reveals an additional difficulty. Self-adaptation contains several strategy parameters, which describe the mutation distribution. These parameters benefits from the recombination (interpolation) of several parent solutions. However, in multi-objective optimization, the individuals of the population converges towards different areas of the Pareto front and efficient strategy parameters differ between the individuals. Thus, recombination will also be discussed for the different selection operators.
In the next section, the multi-objective optimization problem is introduced and we briefly outline the basic concepts of multi-objective evolutionary algorithms, which will also be used later on for comparison. Self-adaptation is presented for multi-objective optimization and the key properties of suitable selection and

recombination operators are discussed. Different multi-objective algorithms are analyzed in terms of their ability to converge to the Pareto front. In addition, the implementation of self-adaptive mutation into these algorithms is discussed. Finally, the performance of the proposed algorithms is analyzed on a 2- and 3-objective test function. In the performance comparison, the number of resulting nondominated solutions of the different algorithms is allowed to be small, since in real-world applications, analyzing these solutions is often an expensive process.

2 Multi-Objective Optimization

2.1 Definition of Multi-Objective Optimization

A multi-objective optimization problem can be formulated by an objective vector $\mathbf{f}$ and a corresponding set of design variables $\mathbf{x}$. Without loss of generality minimization of all objectives is considered:

$$\begin{aligned} &\text{find } \min \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \in F \\ &\quad \text{where } \mathbf{x} = (x_1, x_2, \dots, x_n) \in X, \end{aligned} \tag{1}$$

where $X \in \mathcal{R}^n$ is the n-dimensional design space, $F \in \mathcal{R}^m$ is the m-dimensional objective space. A partial ordering can be applied to solutions to the problem by the dominance criterion. A solution a in X is said to dominate a solution b in X $(a \succ b)$ if it is superior or equal in all objectives and at least superior in one objective. This is expressed as:

$$\begin{aligned} a \succ b, \text{ if } \; &\forall\, i \in \{1, 2, \dots, m\}: \; f_i(a) \leq f_i(b) \;\wedge \\ &\exists\, j \in \{1, 2, \dots, m\}: \; f_j(a) < f_j(b) \end{aligned} \tag{2}$$

The solution a is said to be indifferent to a solution c, if neither solution is dominating the other one. When no a priori preference is defined among the objectives, dominance is the only way to determine, if one solution performs better than the other [16]. The best solutions to a multi-objective problem are the Pareto ideal solutions, which represent the nondominated subset among all feasible solutions. In other words, starting from a Pareto solution, one objective can only be improved at the expense of at least one other objective.

2.2 Self-Adaptation in Multi-objective Optimization

Self-adaptation [10] is associated with mutation or recombination operators and has been mainly used in Evolution Strategies (ES) and Evolutionary Programming (EP) for single objective optimization. In the following we outline the basic principles of self-adapting the mutation distribution of ES as described in [17]. These concepts are subsequently implemented into multi-objective optimization. The optimization of objective functions $\mathbf{f}(\mathbf{x})$, which depend on a set of design variables $\mathbf{x} \in \mathcal{R}^n$ is considered. In ES, mutation is performed by adding a normally distributed random vector to the design variables $\mathbf{x}$:

$$\mathbf{x}' = \mathbf{x} + \mathbf{z}, \quad \mathbf{z} \sim \mathbf{N}(0, \mathbf{C}), \tag{3}$$

where $\mathbf{z}$ is a realization of a normally distributed random vector with zero mean and covariance matrix $\mathbf{C}$. Choosing a constant covariance matrix might be efficient in the beginning of the optimization, but can become inefficient close to the optimum. Adaptation of the mutation distribution has been show to be necessary for efficient optimization algorithms [17].
The elements c_{ij} of the covariance matrix are strategy parameters, which can be build by a set of n standard deviations σ_i and $n\ (n-1)/2$ rotation angles α_k where:

$$\sigma_i^2 = c_{ii} \tag{4}$$

$$\tan(2\alpha_k) = \frac{2c_{ij}}{\sigma_i^2 - \sigma_j^2}, \text{ with } k = \frac{1}{2}(2n-i)(i+1) - 2n + j \tag{5}$$

The mutation of the strategy parameters is performed similarly to the mutation of the design variables by:

$$\sigma_i' = \sigma_i \exp(\tau_0 \mathrm{N}(0,1) + \tau \mathrm{N}_i(0,1)) \tag{6}$$

$$\alpha_k' = \alpha_k + \beta \mathrm{N}_k(0,1), \tag{7}$$

where τ_0, τ and β are the learning rates and recommended values are:

$$\tau_0 = \frac{1}{\sqrt{2n}}, \quad \tau = \frac{1}{\sqrt{2\sqrt{n}}}, \quad \beta = 5^\circ \tag{8}$$

This mutation is referred to as *correlated mutation* or *rotation angle mutation*. Simplifications of the covariance matrix can be obtained in a first step by removing the correlation (i. e., $c_{ij,j\neq i} \equiv 0$) and in a second step by reducing all standard deviations to a single one, i. e., $\sigma_i \equiv \sigma$.
To promote the adaptation of the strategy parameters, the following procedure is recommended [17]:

1. **Non-elitist selection operators should be preferred:**
   Although an elitist strategy ($(\mu + \lambda)$-strategy) guarantees a continuous improvement of the objective value of the parent population, it inhibits the risk of getting stuck in parents with inappropriate strategy parameters with a low chance of generating better offspring. Thus, non-elitist (μ, κ, λ)-strategies with a limited lifetime κ are preferred. A typical population size for parents and offspring are $\mu = 15$ and $\lambda = 100$, respectively [17].
2. **Recombination of a parent population is necessary:**.
   A further improvement is obtained by recombination: The design variables are usually recombined by discrete or intermediate recombination of always two parents and the standard deviations are recombined by computing the mean of all parents (global intermediate recombination). No recombination is usually applied to the rotation angles.

2.3 Multi-Objective Evolutionary Algorithms

We consider multi-objective evolutionary algorithms, performing a population-based search in order to find a set of approximately Pareto-ideal solutions along the Pareto front. Promising methods have been proposed and evaluated by several researchers [1, 4, 18]. The various multi-objective evolutionary algorithms mainly employ a selection operator. In the following we classify different approaches as proposed by Horn [5] and discuss the applicability of these algorithms to self-adaptation. In particular, we focus on the ability of an algorithm to converge with arbitrary precision to the Pareto front.

Independent Sampling: An approximation of the Pareto front can be obtained by performing several independent runs with different aggregation of the objectives by e.g. a weighted sum or a constraint approach. This leads to a discrete approximation of the Pareto front, with each optimization run converging to a different point of the Pareto front. Independent sampling is an ideal candidate for self-adaptation as the multi-objective problem is transformed to a set of single objective problems, thus self-adaptation is directly applicable.
Ranjithan *et al.*[2] proposed to use a constraint method-based evolutionary algorithm (CMEA) for aggregating the objectives. One objective f_h is selected for optimization, while all other objectives $f_{i,i \neq h}$ are treated as constraints:

$$\min f_h, \text{ while } f_i < u_i^t \; \forall \; i = 1, \ldots, m \; ; i \neq h, \tag{9}$$

where u_i^t are the constraint values. For varying the constraint values, different Pareto solutions are obtained.
In order to find appropriate constraint values, the algorithm searches for the extreme corners of the Pareto front by separately optimizing all objectives $f_i, i \neq h$ [5]. Then, for each objective, a certain number of different constraint values u_i^t is chosen uniformly within the extreme corners. For each possible combination of one constraint value per objective, an optimization run is performed and the Pareto front is approximated.
Some knowledge of previous runs can be exploited by using the best solution(s) obtained so far as initial solution(s) for the next run [2].

Cooperative Population Searches with Dominance Criterion: Cooperative population searches converge in a single run towards the Pareto front. The dominance criterion in combination with niching techniques is used in order to select on average the less dominated solutions and preserve diversity in the population, respectively. According to Van Veldhuizen and Lamont [4] this class of fitness assignment is most efficient. Two recent performance comparisons [7] [2] show however that other classes of optimization approaches can also lead to comparable results. One of the most prominent representative is the Strength Pareto Evolutionary Algorithm (SPEA) [1]. SPEA uses the nondominated solutions for the fitness assignment. First, the fitness of each nondominated solution is computed as the fraction of the population, which it dominates. The fitness

of a dominated individual is equal to one plus the fitness of each nondominated solution by which it is dominated. This fitness assignment promotes solutions in sparse areas.
Elitism is a key element in this class of algorithms and represents a technique of preserving always the best solutions obtained so far. It is often performed by preserving the nondominated solutions in an archive. In order to preserve diversity in the archive and to keep its size limited, clustering algorithms are applied. SPEA does not employ a mutation or recombination operator. To compare the performance on continuous problems, Zitzler and Thiele [3] use the polynomial distributed mutation and the simulated binary crossover proposed by Deb *et al.* [19]. Both methods do not implement any adaptation process. So they do not exploit explicitly any knowledge available from the evolution of the population. We discuss these methods in order to analyze their limited convergence towards the Pareto front:
In Fig. 1, a simple example for a 2-objective minimization problem is given with the Pareto front being a straight line between $\mathbf{f} = \{1, 0\}$ and $\mathbf{f} = \{0, 1\}$. For a limited archive size (or number of parents) s, the objective space cannot be completely dominated. For the ideal case, that all archive solutions are uniformly distributed along the Pareto front, the nondominated area is described by the Pareto front and the dashed lines. This area contains, from the aspect of dominance, indifferent solutions, i. e. solutions of the same quality as the considered archive solutions, and the maximal distance of a nondominated solution to the Pareto front can be calculated as $\frac{1}{\sqrt{2}(s-1)}$. Thus, the minimal number of archive solutions in order to dominate a solution with a certain distance to the Pareto front scales with s^{-1} and in order to converge to the Pareto front, an infinite number of archive solutions is necessary. For 3 objectives, it can be shown that the maximal distance of a nondominated solution scales with s^{-2} and the dominance criterion as selection criterion becomes less efficient. The dominance criterion works well for the approximation of the Pareto front, but fails in the final convergence since the archive size of this class of algorithms is usually limited. This is a general problem of selection operators, using the dominance criterion, e. g., SPEA, SPEA2 [3] and NSGA-II [6], as it will be experimentally shown in Section 3.

Cooperative Population Searches without Dominance Criterion: This class of optimization approaches converges towards the Pareto front in a single optimization run, but does not use the dominance criterion within the selection operator. Moreover, it could be considered as performing several Independent Sampling optimizations within a single optimization. Several selections are performed from one population with different aggregation of the objectives. This might be beneficial compared to Independent Sampling, since information can be exchanged while converging as one individual can be preferred by several different aggregations.
One example is the Subdivision Method (SDM) [7], an optimization approach with some similarities to the CMEA. In the following, the algorithm is briefly

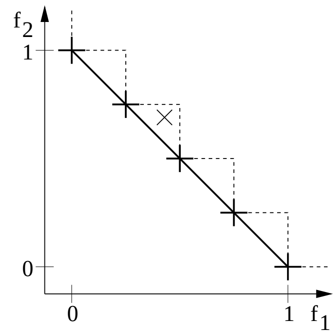

Fig. 1. Disadvantage of the dominance as selection criterion: for a limited number of archive solutions [+ symbols], which may be on the Pareto front [solid line], the objective space cannot be completely dominated, e.g. the solution [x symbol] is not dominated, even though it is still far from the Pareto front.

described and an example for a 2-objective minimization problem is given in Fig. 2. In the objective space, the SDM performs several local (μ, κ, λ) selections and then unifies all selected solutions to the parent population. Similar to the CMEA one objective f_h is selected for optimization, while all other objectives $f_{i,i \neq h}$ are treated as constraints. However, a lower and upper constraint value is set:

$$\min f_h, \ \text{ while } l_i^t \leq f_i \leq u_i^t, \ \forall \ i = 1, \ldots, m \ ; i \neq h. \tag{10}$$

The constraints l_i^t and u_i^t are set such that they divide the objective axis of f_i in k intervals $t = 1, \ldots, k$ of equal width, where the upper constraint u_i^t of an interval t is always the lower constraint l_i^{t+1} of the adjacent interval $t+1$, i. e., $l_i^{t+1} = u_i^t$. The lower constraint value l_i^1 of the first interval and the upper value of the k^{th} interval u_i^k are set equal to the minimal and maximal value for objective i of the current nondominated front, respectively. Thus, the constraints change along the optimization process as the current nondominated front changes. For each possible combination of choosing one interval t for each of the objectives $f_{i,i \neq h}$, a separate selection is performed with respect to f_h, where the constraints are hard, i. e., a solution which violates the constraints is not considered. Then, this process is repeated until each objective is chosen once as a selection criterion, in order to avoid a preference between the objectives. In total $m \cdot k^{m-1}$ local selections are performed.
Self-adaptation can be implemented in this selection operator by the following procedure: The selection process can be considered as performing several local selections in a "subdivided" objective space. The mean distance of the selected individuals to the Pareto front may differ between the local selections, resulting in different sets of efficient strategy parameters. Thus, recombination as described in Sec. 2.2 is always performed within a local selection. Finally, self-adaptive mutation is applied to the recombined individuals.

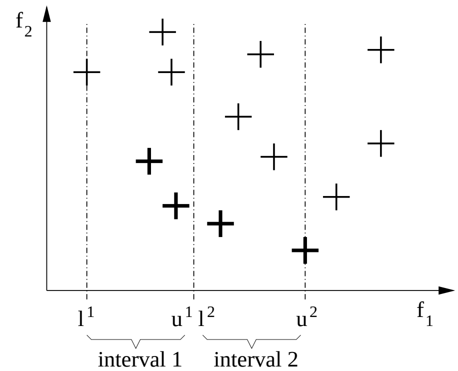

Fig. 2. Selection by SDM for 2 objectives: The objective space is divided along f_1 into two intervals by specifying a hard lower and upper constraint value l and u for f_1, respectively [dash-dotted lines]. From all solutions [+ symbols] in an interval, always the μ best solutions [bold + symbols] with respect to f_2 are selected. Then the procedure is repeated by dividing the space along f_2 and considering f_1 as selection criterion.

3 Experimental Analysis

A simple test function for an arbitrary number of objectives is considered, which is a generalization of the sphere model to multiple objectives [9]. It allows to analyze the convergence of optimization algorithms as a function of the number of objectives:

$$f_i = (x_i - 1)^2 + \sum_{j=1, j \neq i}^{n} x_j^2, \quad i = 1, \ldots, m \ \wedge m \leq n \tag{11}$$

with $x_{1,\ldots,n} \in [-2.0; 2.0]$, i is the index of the objective and the number of variables is set to $n = 10$. For two objectives the Pareto front is given by:

$$x_1 + x_2 = 1, \ x_{3,\ldots,n} = 0 \wedge x_{1,2} \geq 0, \tag{12}$$

and for 3 objectives by:

$$x_1 + x_2 + x_3 = 1, \ x_{4,\ldots,n} = 0 \wedge x_{1,2,3} \geq 0. \tag{13}$$

In the design space, the Pareto fronts of the 2- and 3-objective problem describes a straight line or a plane, respectively.

3.1 Performance Measures

The definition of the quality of the results from a Pareto optimization considers two aspects. The first aspect is the convergence of the solutions towards the Pareto front, and will be addressed by the mean distance D of solutions to the Pareto front. The second aspect reflects the distribution of the solutions, whereas a uniform distribution along the Pareto front is desired. This will be addressed by plotting the objective space.

3.2 Results

In the following we analyze the performance of the 3 different classes of multi-objective algorithms. For simplicity, just one representative of each class is considered in order to focus on the general convergence properties and not on the exact convergence speed.

Independent Sampling: The CMEA is analyzed as a representative of Independent Sampling on a 2-objective sphere. In total 6 independent optimization runs are performed with a $(15, 3, 100)$ strategy, implementing correlated mutation. All initial step sizes are set to 0.1 and 17.000 solutions are evaluated for each run, leading in total to about 100.000 evaluated solutions.
First, 2 single objective optimization runs are started for f_1 and f_2 from random initial solutions. Then, 4 constraint optimization runs are performed, where f_2 is optimized and f_1 is treated as constraint. Since the f_1 value of the best solution of the two single objective runs is about 0 and 2, respectively, the constraints for f_1 for the 4 remaining runs are set to 0.4, 0.8, 1.2, and 1.6.
For the constraint optimization runs, some knowledge of previous runs is exploited by using the best solutions obtained so far as initial solutions for the next run [2]. We consider a hard constraint: If solutions violate the constraint, they are not considered for selection. This constraint ensures a convergence to a point on the Pareto front. A soft penalty with a gradient smaller than the gradient of f_2 may converge to a point in the neighborhood of the Pareto front.
Fig. 3*a* shows the performance measure for all 6 optimization runs, which represents the mean distance of the parent population at each generation to the Pareto front. The figure shows large differences in the convergence between the different runs. While the two single objective optimizations converge linearly, the convergence of the constraint optimizations slows down at a distance of about $D = 0.01$ to the Pareto front. Fig. 4 addresses this aspect and shows the contour lines for f_1 and f_2 and the Pareto front in the (x_1,x_2) space. Each contour line of f_1 represents a different constraint setting. The optimum for each constraint optimization is located in the intersection of a f_1 contour line with the Pareto front. In the vicinity of the optimum, the topology is badly scaled and oriented such that it is difficult to optimize it with the given self-adaptation scheme (compare Hansen *et al.* [20], Fig. 2).
Fig. 7*a* shows the best solution of each optimization run and the Pareto front. While the solutions are equally spaced in f_1 direction, the distribution is nonuniform along the Pareto front, with a concentration on the part of the Pareto front, which is almost parallel to the f_1 axis. This results from treating f_1 and f_2 differently as constraint and objective, respectively.

Cooperative Population Searches with Dominance Criterion: For SPEA, different archive sizes of 30, 100, and 300 are analyzed for the 2- and 3-objective sphere. Here, the focus is on the maximal convergence of a limited archive size to the Pareto front and not on the actual convergence speed of SPEA. The number of parents and offspring is set equal to the archive size and in total 100.000

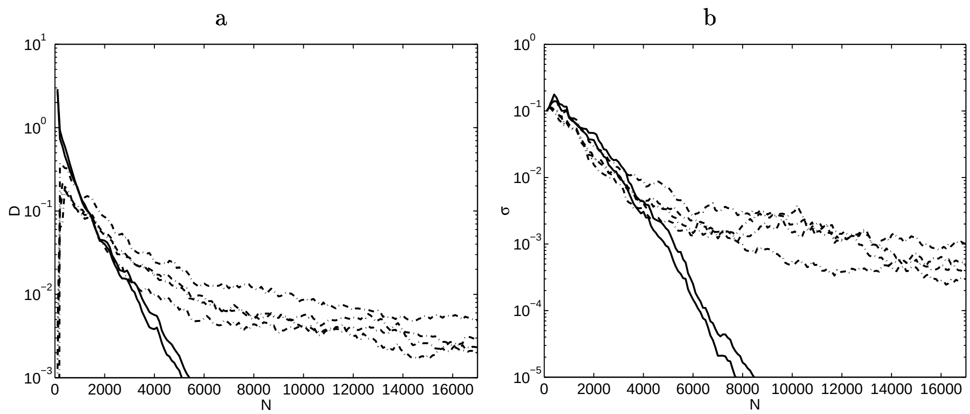

Fig. 3. Mean distance D of the parent population (a) and mean standard deviation σ of the mutation distribution (b) for single objective optimizations [solid lines] and constraint optimizations [dash-dotted lines] for CMEA.

solutions are evaluated. Discrete and intermediate recombination of 2 parents is considered with 33% probability each. In order to analyze just the effect of the archive, a normally distributed mutation with zero mean and standard deviation of 0.0005 and 0.002 is used with a mutation probability of 15% per variable, which lead to the best results.

Fig. 5 shows the convergence measure D, which represents the mean distance of the solutions in the archive to the Pareto front. It can clearly be seen that the convergence stagnates at a certain value of D and this value is decreasing with an increase of the archive size. Comparing the 2- and 3-objective optimization results, a second observation can be made. The final mean distance D for the 3-objective problem is significantly larger than for the 2-objective problem and the increase in the archive size leads to a smaller relative decrease in D. This underlines the previously stated rule that for a limited archive size an evolutionary algorithm, which considers the dominance criterion for selection in a similar way than SPEA, cannot converge to the Pareto front. In addition, in order to obtain the same mean distance D from the Pareto front, the necessary number of archive solutions rises with the number of objectives. The distribution of the solutions of the final archive is shown for the 2-objective problem with an archive size of 300 in Fig. 7*b* and shows the uniform approximation of the Pareto front with a large number of archive solutions.

Cooperative Population Searches without Dominance Criterion: The SDM is now analyzed on the multi-objective sphere. Always 8 solutions are selected in each local selection and a maximal lifetime $\kappa = 3$ is assigned to each solution. The convergence measure D for the SDM is set to the mean distance of all selected parents of a generation to the Pareto front. For the 2-objective problem, 3 intervals are chosen along each objective axis, leading in total to 6 lo-

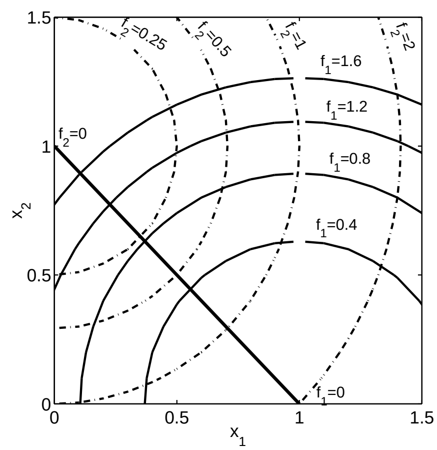

Fig. 4. Convergence difficulty in the CMEA for optimizing f_2 and setting f_1 as constraint. The optimum for a specific constraint value for f_1 is located at the intersection of the Pareto front [bold solid line], with the contour line of f_1 [solid line]. In the vicinity of the optimum, the topology is badly scaled and oriented as shown above.

cal selections and a maximal number of 48 parents. For the 3-objective problem, each objective axis is divided into 2 intervals, leading to a total number of 12 local selections and a maximal number of 96 parents. The bounds of the intervals are obtained from the current nondominated front of the optimization run and thus need no user specification. For self-adaptation, a ratio of 7 offspring per parent is recommended, leading to a total number of 336 and 772 offspring for the 2- and 3-objective problem, respectively. Similar to CMEA, a global intermediate recombination of the parents from one local selection is performed for the variables and step sizes. No recombination is applied to the rotation angles. In total 100.000 solutions are evaluated and the convergence measure is plotted in Fig. 6. In addition, the mean step size of the mutation distribution is given. The convergence speed decreases at a value of about 10^{-2} for the same reason as for the CMEA: The convergence becomes difficult due to the constraint optimization. Here, the convergence speed for the 3-objective problem is about a factor of 2 smaller than for the 2-objective problem. This results mainly due to the doubled number of local selections. The parents of the final population are plotted in Fig. 7*c* and are uniformly distributed along the Pareto front.

4 Conclusions

Evolutionary Algorithms for multi-objective optimization should implement efficient techniques in order to improve convergence towards the Pareto front, while maintaining diversity spanning the front. We study three different classes of multi-objective algorithms and compare a representative of each class with

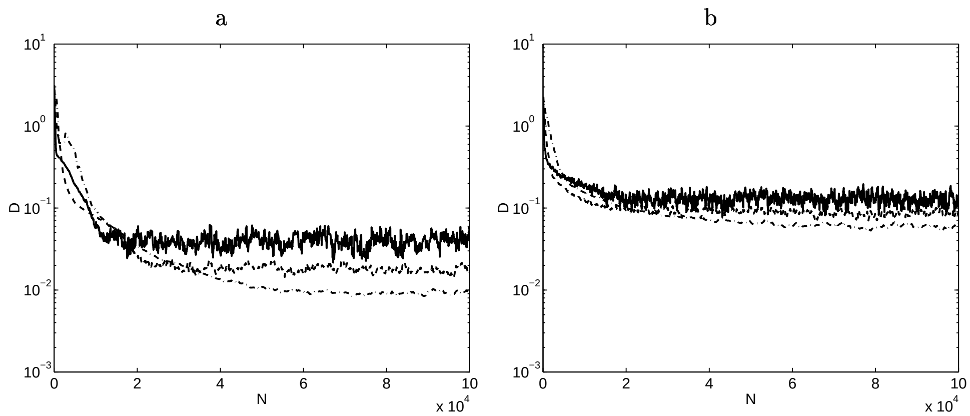

Fig. 5. Mean distance D of the archive solutions of SPEA for the 2-objective (a) and 3-objective (b) sphere problem. The archive sizes is varied between 30 [solid line], 100 [dashed line] and 300 [dash-dotted line].

each other. The question is which of these algorithms is able to converge to the Pareto front with an arbitrary precision.

We found that cooperative population searches like SPEA, which use the dominance criterion in the fitness assignment, cannot approximate the Pareto front with arbitrary precision. For a 2-objective optimization problem, the necessary number of archive solutions to assign fitness by dominance scales inversely with the distance of solutions to the Pareto front. For 3 objectives, convergence becomes even more difficult, since the necessary number of archive solutions scales inversely with the square of the distance. This result holds for other algorithms using the dominance criterion and a limited population (e. g. NSGA-II, SPEA2).

The algorithms CMEA and SDM do not use dominance. In these algorithms, one objective is selected for optimization, while the other objectives are treated as constraints. Both algorithms converge to a fixed number of discrete points on the Pareto font, which can be reached in arbitrary precision. For the considered test functions, this number is significantly smaller than the number of nondominated solutions of SPEA. However, the limited number of converged solutions of CMEA and SDM is often sufficient in real-world applications, especially if the analysis of these solutions is expensive. Thus, algorithms like CMEA and SDM are interesting alternatives to the well established algorithms based on the dominance criterion.

CMEA finds one optimal point in each optimization run and thus needs to be run for several times in order to find an approximation of the Pareto front. In contrast, SDM finds an approximate Pareto front in a single optimization run. It is shown that self-adaptation can easily be applied to CMEA and SDM and that both algorithms converge successfully to the Pareto front and outperform SPEA in terms of the final distance to the Pareto front. Comparing CMEA and SDM in terms of convergence speed, CMEA is faster on the considered test func-

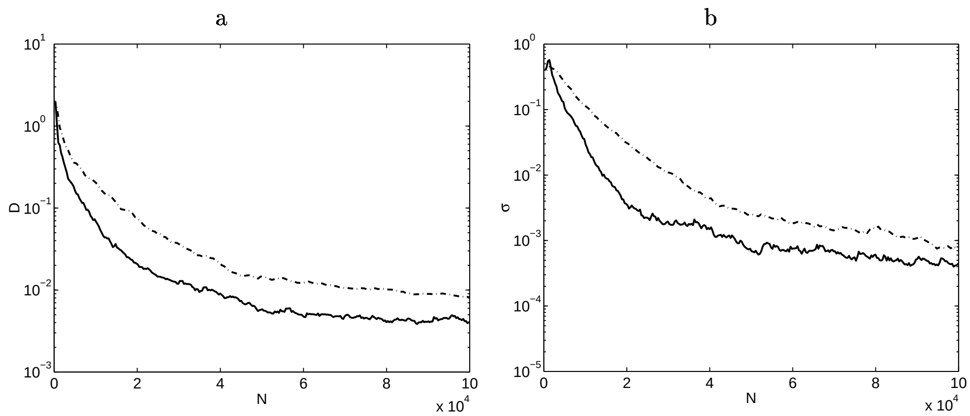

Fig. 6. Convergence of the SDM for the 2-objective [solid line] and 3-objective [dash-dotted line] sphere problem. The mean distance of the parents to the Pareto front is given (a) as well as the mean standard deviation for the mutation (b).

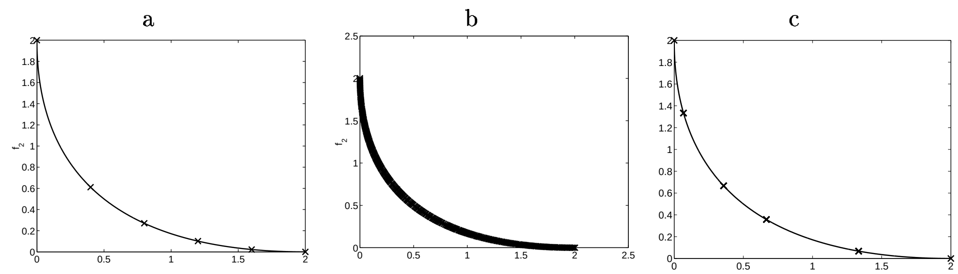

Fig. 7. Location of the best solution from each independent run of CMEA (a), the final archive of SPEA (b) and final parent population of SDM (c) [x] on the Pareto front [bold line].

tion, although it is not known if this generalizes to other functions and to other adaptation schemes. For CMEA, one has to decide before optimization, which objective is optimized subject to the other objectives, which are then treated as constraints. This is in contrast to SDM that gives no a-priori preference to any of the objectives.

Some difficulty in converging has been found for CMEA and SDM. CMEA and SDM transfer the multi-objective sphere problem into a difficult constraint optimization problem: While optimizing a single objective of the sphere problem converges linearly (see Fig. $3a$, solid lines), the convergence speed of the constraint optimization problem decreases over the number of function evaluations (see Fig. $3a$, dash-dotted lines). In general the question arises if objectives could be aggregated by a different method, leading to a linear convergence also in the constraint case. In addition, Hansen *et al.* [20] stated some difficulty of self-adaptation in generating arbitrary correlation distributions and thus methods like the Covariance Matrix Adaptation [20] might perform better on the constraint problem.

5 Acknowledgments

We would like to acknowledge support from the Commission for Technology and Innovation, Switzerland (Project 4817.1) and Alstom Power Technology Center Dättwil, Switzerland.

References

1. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation **3** (1999) 257–271
2. Ranjithan, S.R., Chetan, S.K., Dakshima, H.K.: Constraint Method-Based Evolutionary Algorithm (CMEA) for Multiobjective Optimization. In Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D., eds.: First International Conference on Evolutionary Multi-Criterion Optimization, Springer-Verlag. Lecture Notes in Computer Science No. 1993 (2001) 299–313
3. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland (2001)
4. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. Evolutionary Computation **8** (2000) 125–147
5. Horn, J.: Multicriterion decision making. In Bäck, T., Fogel, D.B., Michalewicz, Z., eds.: Handbook of Evolutionary Computation, Sec. F1.9: pp. 1-15. (1997)
6. Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H.P., eds.: Proceedings of the Parallel Problem Solving from Nature VI Conference, Paris, France, Springer. Lecture Notes in Computer Science No. 1917 (2000) 849–858
7. Büche, D., Dornberger, R.: New evolutionary algorithm for multi-objective optimization and the application to engineering design problems. In: Proceedings of the Fourth World Congress of Structural and Multidisciplinary Optimization, Dalian, China, International Society of Structural and Multidisciplinary Optimization (ISSMO) (2001)
8. Kursawe, F.: A Variant of Evolution Strategies for Vector Optimization. In Schwefel, H.P., Männer, R., eds.: Parallel Problem Solving from Nature. 1st Workshop, PPSN I. Volume 496 of Lecture Notes in Computer Science., Berlin, Germany, Springer-Verlag (1991) 193–197
9. Laumanns, M., Rudolph, G., Schwefel, H.P.: Mutation Control and Convergence in Evolutionary Multi-Objective Optimization. In: Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001), Brno, Czech Republic (2001)
10. Schwefel, H.P.: Numerical Optimization of Computer Models. Wiley, Chichester (1981)

11. Sbalzarini, I.F., Müller, S., Koumoutsakos, P.: Microchannel optimization using multiobjective evolution strategies. In Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D., eds.: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO), Zürich, Switzerland, Springer Lecture Notes in Computer Science (2001) 516–530
12. Abbass, H.A.: The Self-Adaptive Pareto Differential Evolution Algorithm. In: Congress on Evolutionary Computation (CEC'2002). Volume 1., Piscataway, New Jersey, IEEE Service Center (2002) 831–836
13. Storn, R., Price, K.: Minimizing the real functions of the icec'96 contest by differential evolution. In: IEEE Conference on Evolutionary Computation. (1996) 842–844
14. Büche, D., Milano, M., Koumoutsakos, P.: Self-Organizing Maps for Multi-Objective Optimization. In Barry, A.M., ed.: GECCO 2002: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference, New York, AAAI (2002) 152–155
15. Kohonen, T.: Self-Organizing Maps. 3 edn. Springer series in information sciences (2001)
16. Fonseca, M.C., Fleming, P.J.: Multi-objective genetic algorithms made easy: Selection, sharing and mating restrictions. In: Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Application. (1995) 45–52
17. Bäck, T.: Self-adaptation. In Bäck, T., Fogel, D.B., Michalewicz, Z., eds.: Handbook of Evolutionary Computation, Sec. C7.1: pp. 1-15. (1997)
18. Coello Coello, C.A.: An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In: Congress on Evolutionary Computation. (1999) 3–13
19. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Systems **9** (1995) 115–148
20. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9** (2001) 159–195