Adaptive Variance Scaling in Continuous Multi-Objective Estimation-of-Distribution Algorithms

Peter A.N. Bosman
Centre for Mathematics and Computer Science
P.O. Box 94079
1090 GB Amsterdam
The Netherlands
Peter.Bosman@cwi.nl

Dirk Thierens
Utrecht University, Institute of Information and Computing Sciences
P.O. Box 80089
3508 TB Utrecht
The Netherlands
Dirk.Thierens@cs.uu.nl

ABSTRACT

Recent research into single–objective continuous Estimation–of–Distribution Algorithms (EDAs) has shown that when maximum–likelihood estimations are used for parametric distributions such as the normal distribution, the EDA can easily suffer from premature convergence. In this paper we argue that the same holds for multi–objective optimization. Our aim in this paper is to transfer a solution called Adaptive Variance Scaling (AVS) from the single–objective case to the multi–objective case. To this end, we zoom in on an existing EDA for continuous multi–objective optimization, the MIDEA, which employs mixture distributions. We propose a means to combine AVS with the normal mixture distribution, as opposed to the single normal distribution for which AVS was introduced. In addition, we improve the AVS scheme using the Standard–Deviation Ratio (SDR) trigger. Intuitively put, variance scaling is triggered by the SDR trigger only if improvements are found to be far away from the mean. For the multi–objective case, this addition is important to keep the variance from being scaled to excessively large values. From experiments performed on five well–known benchmark problems, the addition of SDR and AVS is found to enlarge the class of problems that continuous multi–objective EDAs can solve reliably.

Categories and Subject Descriptors

G.1 [**Numerical Analysis**]: Optimization; I.2 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Performance, Experimentation

Keywords

Evolutionary Algorithms, Estimation of Distribution Algorithms, Multi–Objective Optimization, Adaptive Variance Scaling

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

1. INTRODUCTION

Estimation–of–distribution algorithms (EDAs, [16, 19, 23]) are a class of evolutionary algorithms in which the main operator of variation is the estimation of a probability distribution from the selected solutions and the subsequent sampling from the estimated distribution to generate new solutions, i.e. offspring. EDAs attempt to induce and exploit structure from the optimization problem. The probability distribution constitutes an explicit, probabilistic, search bias.

In general, for any optimization algorithm to be successful when solving a certain optimization problem, the structure of the problem needs to match the bias of the optimization algorithm. Recent studies have shown that the EDA approach in continuous spaces, specifically when maximum–likelihood normal distributions are used, is not always successful [4, 12]. Also, it has been pointed out more clearly under which conditions an EDA *is* expected to be successful [11]. Summarizing, it is required that the structure of the problem can be modeled by the probability distribution and the estimation procedure can do this modeling well. For the normal distribution however, this not always the case.

As the normal distribution itself is a single peak, it can match the contour–lines of a single peak in the fitness landscape. Things are different for slope–like situations, i.e. when the optimum is outside the range of selected solutions. The true structure is then misrepresented by a maximum–likelihood estimate because the normal distribution focuses search around its mean. Relying the search on maximum–likelihood estimates potentially misleads the EDA and can therefore cause premature convergence.

Although there is currently no theoretical justification for the same problem arising in the multi–objective case, it is not hard to see that similar problems may indeed arise. One might argue that the variance is "artificially" kept nonzero as upon convergence there will be multiple rank–0 solutions with different configurations in the parameter space. However, this non–zero variance may only be related to a certain subspace of the parameter space. The variance in the directions in which the entire Pareto front may be advanced can vanish, making progression very slow or even non–existing.

Recently, a technique was introduced to remedy the problem of the prematurely vanishing variance with promising results [11]. In this paper, we discuss transferring this technique, called adaptive variance scaling (AVS), to the multi–objective case. The addition of variance scaling brings about

a different view on the way model–based search is performed with continuous EDAs. Originally, the covariance matrix was estimated using maximum likelihood directly from data (i.e. the selected solutions). With variance scaling, the covariance matrix is adapted according to additional sources of information. EDAs are not the only approach to model–guided search. Specifically, when regarding the use of the normal distribution, there are clear similarities with evolution strategies (ES) [2], or more recently, the CMA–ES [13, 14]. But also approaches like particle–swarm optimization (PSO) share a similar notion of maintaining, updating and adapting a model during search. All approaches have a different, but solid, rationale and background. An advantage of the EDA approach that also holds for the AVS extension is that it is conceptually easy to understand. Its choices are well motivated and principled. In addition, the building of mixture distributions by clustering the objective space in multi–objective optimization, was first proposed in an EDA approach, i.e. the multi–objective mixture–based iterated density–estimation evolutionary algorithm (MIDEA) [6, 25]. It has been shown that the use of mixture probability distributions leads to better results. As such, it is important to investigate and advance all these model–based approaches and specifically EDAs, on which this paper is focused.

Because AVS was designed only with the single normal distribution in mind, we will have to expand this technique to the level of mixtures of normal distributions. The combination of mixture distributions with AVS to reduce the risk of premature convergence is an important next step in the development of continuous multi–objective estimation–of–distribution algorithms (MOEDAs) that can reliably tackle a wide range of problem difficulty.

The remainder of this paper is organized as follows. First, we briefly discuss mixture distributions in MOEDAs as well as the specific MOEDA called MIDEA in Section 2. Then, in Section 3 we briefly recall AVS for single–objective optimization and propose the extension of AVS to normal mixture distributions. In Section 4 we improve the AVS approach further by introducing the standard–deviation ratio (SDR) trigger. We test the performance of the MIDEA with and without AVS and SDR in Section 5 on five benchmark problems. We present our final conclusions in Section 6.

2. MULTI–OBJECTIVE EDAS

2.1 Mixture distributions, clusters and multiple objectives

A mixture probability distribution is a weighted sum of k probability distributions. Let $\boldsymbol{X}$ be the random variable that represents the entire space of solutions to the problem at hand. Typically, the solutions are fixed–length vectors that constitute a Cartesian search space, making $\boldsymbol{X}$ a fixed–length vector of random variables X_j where X_j is associated with the j–th problem variable. A mixture probability distribution can now be defined as:

$$P^{mixture}(\boldsymbol{X}) = \sum_{i=0}^{k-1} \beta_i P^i(\boldsymbol{X}) \quad (1)$$

where $\beta_i > 0$, $i \in \{0, 1, \ldots, k-1\}$ and $\sum_{i=0}^{k-1} \beta_i = 1$. The β_i are called the mixing coefficients and each probability distribution P^i is called a mixture component.

The power of mixture distributions mainly lies in the combination of multiple, typically simpler, distributions. In this way, accurate descriptions can be obtained of the data in different parts of the sample space. Although the mixture components may overlap heavily, it is often more convenient to think of them as a means of clustering the sample space. One mixture component then represents one cluster, making the mixture components spatially separated [4, 22, 25].

For multi–objective optimization, spatial separation renders mixture distributions particularly useful [6, 25]. Using clustering in the objective space, the distributions related to these clusters can portray specific information about the different regions along the Pareto front. This typically increases the effectiveness in advancing the Pareto front. Each distribution needs only to focus on moving solutions in a specific part of the search space. A parallel exploration along the Pareto front is thereby obtained that may very well provide a better spread of new solutions along the Pareto front than when a single non–mixture distribution is used.

2.2 The MIDEA

The MIDEA framework is a framework for multi–objective optimization with EDAs [6, 25]. Components that are specific for this framework are the way in which variation, selection and replacement are performed. A coarse–grained outline of the framework is given in Figure 1.

MIDEA
1 Initialize a population of n (random) solutions
2 Iterate until termination
2.1 Select the best $\lfloor \tau n \rfloor$ solutions
2.2 Generate $n - \lfloor \tau n \rfloor$ new solutions by variation: estimate a mixture distribution from the selected solutions, then draw samples from it.
2.3 Replace the non–selected solutions with the new solutions

Figure 1: Outline of the MIDEA framework.

2.2.1 Variation

The probability distribution used in the MIDEA is the mixture distribution as described in Section 2.1. Each mixture component is assigned an equally large mixing coefficient, i.e. $\beta_i = 1/k$. This is done so as to distribute the solutions as good as possible along the Pareto front. Giving each cluster an equal probability of producing new solutions maximizes parallel exploration along the Pareto front [6].

There are various ways to estimate mixture distributions. Currently existing implementations employ clustering in the objective space to partition the selected solutions into subsets. For each subset, a probability distribution is then estimated. Using partitioning, specific emphasis is put on the spatial separation of the mixture components. Each mixture component is a factorized distribution [18], similar to the approach adopted by BOA, FDA and IDEA [4, 20, 23].

2.2.2 Selection and replacement

The motivation behind truncation selection and replacement of all non–selected solutions in the MIDEA framework is that of elitism. The currently–best solutions are always preserved from one generation to the next. Elitism has proven to be advantageous if the variation operator is capable of effectively exploiting problem structure [24]. As finding a competent means of variation is exactly the goal of EDA research, the choice of elitism is natural.

Two common ways to compute a ranking to use in truncation selection in the multi–objective case are non–dominated ranks [8, 10] and domination count [9, 25]. In MIDEA the latter is often chosen although it has been shown that there

is not much difference between the two approaches in practice [5]. The domination count of a solution is the number of times it is dominated by another solution in the population. The lower the rank, the better the solution.

Whenever there are more solutions of rank 0 than the number of solutions to be selected, diversity is used to make the actual selection from all available rank–0 solutions. Note that this situation can easily happen if the search space is continuous. Then, there is typically an infinite number of solutions in the Pareto–optimal front. A nearest–neighbour heuristic is used with Euclidean distances measured in the objective space [5]. Each dimension is scaled linearly by its observed range to take into account different scales of objectives. For each rank–0 solution, the nearest neighbour in the set of selected solutions is stored and updated during selection. Until enough solutions are selected, the solution with the largest nearest–neighbour distance is selected. By selecting the first solution to be maximal in a randomly chosen objective, selection automatically attempts to divide up the space between selected solutions evenly.

3. ADAPTIVE VARIANCE SCALING

3.1 AVS and the normal distribution

To remedy the problem of the prematurely vanishing variance, the variance can be scaled. This was first noted only recently [21]. One successful scheme for doing variance scaling in an adaptive fashion (i.e. during optimization) was recently introduced under the name adaptive variance scaling (AVS) [11]. This scheme significantly improves performance in the single–objective case and allows the EDA to solve problems that it couldn't solve without scaling the variance. We now briefly summarize AVS.

The smaller the variance, the smaller the area of exploration for the EDA. The variance in the normal distribution is stored in the covariance matrix $\mathbf{\Sigma}$. A variance multiplier c^{AVS} is maintained. Upon sampling new solutions, the distribution is scaled by c^{AVS}, i.e. the covariance matrix used for sampling is $c^{\text{AVS}}\mathbf{\Sigma}$ instead of just $\mathbf{\Sigma}$. If the best fitness value improves in one generation, then the current size of the variance allows for progress. Hence, a further enlargement of the variance may allow further improvement in the next generation. To fight the variance–diminishing effect of selection, the size of c^{AVS} is scaled by $\eta^{\text{INC}} > 1$. If on the other hand the best fitness does not improve, the range of exploration may be too large to be effective and the variance multiplier should be decreased by a factor $\eta^{\text{DEC}} \in [0, 1]$. For symmetry, $\eta^{\text{DEC}} = 1/\eta^{\text{INC}}$. As the objective of the AVS scheme is to enlarge the variance to prevent premature convergence, c^{AVS} is not allowed to become smaller than 1.

3.2 AVS and the normal mixture distribution

In the $\mathbb{M}$IDE$\mathbb{A}$, a mixture of normal distributions is used instead of a single normal distribution. Here we describe an extension of AVS to a spatially separated set of clusters to be used in the normal mixture distribution.

3.2.1 Clustering

Each distribution explores its own region. Hence it makes sense to assign each distribution P^i a different $c^{\text{AVS},i}$. This can however not be combined directly with clustering. There is not necessarily a clear correspondence between the clusters found in generation j and the clusters found in generation $j + 1$. This correspondence is however important for the progression of the $c^{\text{AVS},i}$. For this reason, we propose an alternative approach to building the mixture distribution.

The number of mixture components k is fixed beforehand. Each normal distribution is assigned its own subpopulation of equal size n^{subpop}, i.e. $n = kn^{subpop}$. During optimization, the subpopulations are kept spatially separated, i.e. a clustering is maintained rather than recomputed in each generation anew. Enforcing spatial separation in the different MOEDA–phases is described in subsections 3.2.2– 3.2.4. Because now there is a clear correspondence of the normal distributions between two subsequent generations, AVS can be used for each subpopulation separately.

3.2.2 Initialization

Initially, n random solutions are generated. Subsequently, they are divided into k subpopulations of size n^{subpop}. To separate them spatially, an approach similar to the diversity–selection approach in selection is used. First, k solutions are selected in the same way as in selection when selecting from rank–0 solutions only. These solutions are called the leaders of the subpopulations. Then, each subpopulation, in turn, is expanded with the solution that is closest to its leader. This process is repeated until all solutions have been assigned.

3.2.3 Selection

To obtain a joint effort of the subpopulations to move toward the Pareto–optimal front, selection is performed on the union of all subpopulations (i.e. the entire population). Each solution is marked with its subpopulation of origin to identify the selected solutions in each subpopulation. The estimation of the distributions in this way is done from spatially separated clusters. It may happen though, that no solutions are selected from subpopulation i. In this case, $c^{\text{AVS},i}$ is reset to 1 and the distribution is cloned from another subpopulation that does not have an empty set of selected solutions. Because over subsequent generations, spatial separation is enforced, these populations will drift apart again, spreading the search effort along the Pareto front.

3.2.4 Offspring generation and replacement

New solutions are generated by drawing a single new solution from each distribution in turn. This process is repeated until $n - \lfloor \tau n \rfloor$ solutions have been generated. To enforce spatial separation, a solution that was generated from the i–th distribution is not by default assigned to the i–th subpopulation. Instead, the solution is assigned to the subpopulation to which it is nearest and doesn't already contain n^{subpop} solutions. The distance of a solution to a subpopulation is taken to be the average Euclidean distance (linearly scaled by the observed ranges of the objectives) between that solution and all members of the subpopulation. If a subpopulation has filled up to n^{subpop} solutions, it is no longer allowed to generate new solutions.

3.3 AVS, multiple objectives and convergence

The AVS scheme is based upon whether or not improvements are found when making new solutions. The notion of improvement that we use for the multi–objective case is a straightforward extension of the single–objective case: an improvement is obtained for subpopulation i if any new solution in subpopulation i dominates any elitist solution.

Because we are dealing with continuous objectives, some undesirable convergence properties can be expected with

AVS. Without a technique such as ε–dominance archiving, true convergence to the Pareto–optimal front may not occur [17]. As soon as selection based on diversity is required to prune superfluous rank–0 solutions, it is possible that over multiple generations solutions end up in the population that are dominated by solutions that were pruned earlier. Hence, maintaining the best solutions of the current generation doesn't lead to true elitism. This also happens with the selection method employed here. Improvements thus keep happening, even when no actual improvement is made in terms of convergence toward the Pareto–optimal front. For AVS this means that the variance is scaled up even in the convergence phase, which worsens the oscillatory behavior.

To overcome this, we propose to maintain an external, elitist, archive that is updated in a fashion similar to ε–dominance [17]. This archive is only used to have a better notion of improvement. The archive consists of all the best solutions seen so far. Because the objectives are continuous, there are usually infinitely many non–dominated solutions possible. To prevent the archive from growing to an extreme size, the search space is discretized into hypercubes by discretizing each objective separately. Only one solution per hypercube is allowed to be in the archive. Once the offspring are generated, they are compared to the solutions in the archive. If the offspring is dominated by any archive solution, it is not entered into the archive. If the offspring is not dominated, it is added to the archive if and only if the hypercube that it resides in does not already have a representative solution in the archive. Finally, when a new solution is entered into the archive, all solutions in the archive that are dominated by it, are removed. A new solution now is said to be an improvement if and only if it is added to the archive. As the external archive now is truly elitist over all generations, this definition of improvement does not lead to additional problematic scalings upon convergence.

4. STANDARD–DEVIATION RATIO (SDR) TRIGGER

In the AVS scheme, improvements automatically increase c^{AVS}. Improvements however do not always mean that the variance needs to be enlarged. This is especially the case if the mean is near the optimum. In this case, the induced bias of the normal pdf already leads the EDA to the optimum.

In multi–objective optimization, this problem plays an important role. Many improvements are made every generation as the Pareto–front is advanced. Because an improvement can be made anywhere along the front, the probability of obtaining at least one improvement in a single generation is much larger than in the single–objective case. Many of these improvements are however likely to be close to the means of the distributions along the front. Without SDR, the variance multipliers will grow very large, making further advancement much slower. Increasing the variance will then only slow down convergence, as the EDA is forced to explore a larger area of the search space unnecessarily.

If improvements mostly take place far away from the mean, then obviously, the mean needs to shift. As we know that mean–shift is problematic for maximum–likelihood normal EDAs, this is a situation in which AVS is called for. If however most of the improvements are obtained near the mean, then the EDA with maximum–likelihood parameters already has a good focus and no further variance enlargement is required. It is known (see, e.g. [1]) that for any value of the standard deviation σ, a fixed percentage of the density of the normal distribution is contained within $[\mu - c\sigma, \mu + c\sigma]$ where μ is the mean of the normal distribution and $c \geq 0$. Now, let $\overline{\boldsymbol{x}^{\text{IMP},i}}(t)$ denote the average of improvements in generation t for subpopulation i. We propose to trigger the further enlargement of the variance multiplier of subpopulation i in generation $t+1$ whenever $\overline{\boldsymbol{x}^{\text{IMP},i}}(t)$ lies further away from the estimated mean than a single standard deviation, i.e. outside the $\approx 68\%$ region surrounding the mean. This amounts to computing the ratio of the distance of $\overline{\boldsymbol{x}^{\text{IMP},i}}(t)$ to the mean and the distance of the contour line of one standard deviation to the mean in the same direction. We call this ratio the standard–deviation ratio (SDR). Note that this trigger is independent of the sample range and has a fixed, predefined notion of being "close" to the mean. The SDR–AVS–MIDEA is summarized in pseudo–code in Figure 2.

SDR–AVS–MIDEA

1 Generate n (random) solutions
2 Evaluate the objectives of all solutions
3 Choose k far–apart leaders with the nearest–neighbour heuristic
4 Repeat until all solutions are assigned
  4.1 For $i \in \{0, 1, \ldots, k-1\}$
    4.1.1 Assign to subpopulation i the nearest solution
5 $c^{\text{AVS},i} \leftarrow 1, \quad i \in \{0, 1, \ldots, k-1\}$
6 Iterate until termination
  6.1 Select the best $\lfloor \tau n \rfloor$ solutions from all subpopulations
  6.2 Assign selected solutions back to subpopulations of origin
  6.3 Estimate a normal distribution for each subpopulation
  6.4 Scale the covariance matrices, i.e. $\boldsymbol{\Sigma}^i \leftarrow c^{\text{AVS},i}\boldsymbol{\Sigma}^i$
  6.5 For each subpopulation i with 0 selected solutions
    6.5.1 Copy distribution parameters from a randomly chosen subpopulation with > 1 selected solutions and reset $c^{\text{AVS},i} \leftarrow 1$
  6.6 $n^{\text{IMP},i} \leftarrow 0$, $\overline{\boldsymbol{x}^{\text{IMP},i}} \leftarrow (0, 0, \ldots, 0), \quad i \in \{0, 1, \ldots, k-1\}$
  6.7 $i \leftarrow 0$
  6.8 Repeat $n - \lfloor \tau n \rfloor$ times
    6.8.1 If subpopulation i not yet full
      6.8.1.1 Generate new solution $\boldsymbol{o}$ from distribution i
      6.8.1.2 Evaluate the objectives of solution $\boldsymbol{o}$
    6.8.2 Assign $\boldsymbol{o}$ to its nearest, non–full, subpopulation
    6.8.3 If $\boldsymbol{o}$ is an improvement
      6.8.3.1 $n^{\text{IMP},nearest} \leftarrow n^{\text{IMP},nearest} + 1$
      6.8.3.2 $\overline{\boldsymbol{x}^{\text{IMP},nearest}} \leftarrow \overline{\boldsymbol{x}^{\text{IMP},nearest}} + \boldsymbol{o}$
    6.8.4 $i \leftarrow (i+1) \bmod k$
  6.9 For each subpopulation $i \in \{0, 1, \ldots, k-1\}$
    6.9.1 If $n^{\text{IMP},i} > 0$
      6.9.1.1 $\overline{\boldsymbol{x}^{\text{IMP},i}} \leftarrow \overline{\boldsymbol{x}^{\text{IMP},i}}/n^{\text{IMP},i}$
      6.9.1.2 Compute SDR from $\overline{\boldsymbol{x}^{\text{IMP},i}}$
      6.9.1.3 If SDR > 1
        6.9.1.3.1 $c^{\text{AVS},i} \leftarrow c^{\text{AVS},i}\eta^{\text{INC}}$
      else
        6.9.1.2 $c^{\text{AVS},i} \leftarrow c^{\text{AVS},i}\eta^{\text{DEC}}$
    6.9.2 If $c^{\text{AVS},i} < 1$
      6.9.2.1 $c^{\text{AVS},i} \leftarrow 1$

Figure 2: Standard–Deviation Ratio (SDR) triggering and Adaptive Variance Scaling (AVS) in the MIDEA. Gray lines are SDR–only.

5. EXPERIMENTS

5.1 Setup

5.1.1 Benchmark problems

We used the well–known problems EC_i, $i \in \{1, 2, 3\}$. For specific details regarding the difficulty of these problems we refer the interested reader to the literature [7, 26]. Their definitions are presented in Table 1. We have taken two more problems from more recent literature on numerical multi–objective optimization [3]. These problems are labeled BD_i,

$i \in \{1, 2\}$. These problems were introduced to remedy a shortcoming in the range of problem–difficulties presented by the EC_i problems. Both problems make use of Rosenbrock's function. Premature convergence on this function is likely without proper induction of the structure of the search space. In the single–objective case for instance, EDAs based on a single maximum–likelihood normal distribution cannot optimize Rosenbrock's function efficiently, especially as the dimensionality of the problem increases. Function BD_2 is harder than BD_1 in the sense that the objective functions overlap in all variables instead of only in the first one. Finally, we have scaled the objectives of BD_2 to ensure that the optimum of all problems is in approximately the same range. By doing so, using the same value to reach for the $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F} \to \mathcal{S}}$ indicator on all problems corresponds to a similar required front–quality on all problems.

None of the problems in our test–suite have locally optimal fronts. This allows us to analyze more clearly the convergence properties of the MOEDAs at hand.

Name	Objectives	Domain
BD_1	$f_0 = x_0$ $f_1 = 1 - x_0 + \gamma$ $\gamma = \sum_{i=1}^{l-2} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right)$	$[0; 1]\times$ $[-5.12; 5.12]^9$ $(l = 10)$
BD_2^s	$f_0 = \frac{1}{l} \sum_{i=0}^{l-1} x_i^2$ $f_1 = \frac{1}{l-1} \sum_{i=0}^{l-2} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right)$	$[-5.12, 5.12]^{10}$ $(l = 10)$
EC_1	$f_0 = x_0, \quad f_1 = \gamma \left(1 - \sqrt{f_0/\gamma}\right)$ $\gamma = 1 + 9 \left(\sum_{i=1}^{l-1} x_i/(l-1)\right)$	$[0, 1]^{30}$ $(l = 30)$
EC_2	$f_0 = x_0, \quad f_1 = \gamma \left(1 - (f_0/\gamma)^2\right)$ $\gamma = 1 + 9 \left(\sum_{i=1}^{l-1} x_i/(l-1)\right)$	$[0, 1]^{30}$ $(l = 30)$
EC_3	$f_0 = x_0$ $f_1 = \gamma \left(1 - \sqrt{f_0/\gamma} - (f_0/\gamma)\sin(10\pi f_0)\right)$ $\gamma = 1 + 9 \left(\sum_{i=1}^{l-1} x_i/(l-1)\right)$	$[0, 1]^{30}$ $(l = 30)$

Table 1: The benchmark problems used for testing.

5.2 Dominance and continuous objectives

Dominance, i.e. whether solution $\boldsymbol{x}^0$ is better than solution $\boldsymbol{x}^1$, is defined as (minimization):

$$\boldsymbol{x}^0 \succ \boldsymbol{x}^1 \Leftrightarrow \left(\forall i : f_i(\boldsymbol{x}^0) \leq f_i(\boldsymbol{x}^1)\right) \wedge \left(\exists i : f_i(\boldsymbol{x}^0) < f_i(\boldsymbol{x}^1)\right) \quad (2)$$

Directly using equation 2 in the selection operator may however lead to unwanted behavior, especially in the case of continuous objectives. There may for instance be many solutions that are extremely close to each other in objective i without any solution dominating another solution. Especially if the problem exhibits structure that makes such a situation likely (because for instance there is a higher density around a certain value for objective i), these structures may get focused on by the optimizer. In terms of dominance, there is no distinguishing between keeping solutions farther apart or having them all at a close distance. This is all the more likely to happen if (double–precision) real values are used instead of a discretization of the search space.

To prevent this problem from occurring, we slightly change the dominance criterion. If for any objective i, $f_i(\boldsymbol{x}^0)$ and $f_i(\boldsymbol{x}^1)$ are closer to each other than some threshold $\theta^{distance}$, i.e. $|f_i(\boldsymbol{x}^0) - f_i(\boldsymbol{x}^1)| < \theta^{distance}$, that objective is dropped from the dominance relation. This means that from all solutions nearer to each other than $\theta^{distance}$ in objective i, the solution that dominates in the other objectives, is the best. This is quite similar to the concept of ε–dominance [17]. However, this specific remedy is only meant to prevent unwanted behavior due to a real–valued representation and a real–valued search space and not to optimize convergence.

5.3 General algorithmic setup

For selection we set τ to 0.3, conforming to earlier work [3, 6] and the rule–of–thumb for FDA [20]. For AVS, we set $\eta^{\text{DEC}} = 0.9$ and $c^{\text{AVS–MAX}} = 10.0$, conforming to the existing literature [11]. We further set $\theta^{distance} = 10^{-5}$ and the discretization of the objectives for the external archive to 10^{-3}. These values are small enough to prevent the unwanted dominance behavior as mentioned in Section 5.2 but also still allow many more rank–0 solutions to exist than the number of solutions we will have in our population for testing. All results are averaged over 100 independent runs.

If the variables move outside their bounded ranges, some objective values can become non–existent. It is therefore important to keep the variables within their ranges. A simple repair mechanism that changes a variable to its boundary value if it has exceeded this boundary value gives artifacts that may lead to false conclusions about the performance of the tested MOEDAs. We have therefore adapted the sampling procedure to reject all solutions that are out of bounds.

5.3.1 Measuring performance

Performance is measured using the non–dominated solutions in the population upon termination. We call such a subset an approximation set and denote it by $\mathcal{S}$. A performance indicator is a function of approximation sets $\mathcal{S}$ and returns a real value that indicates how good $\mathcal{S}$ is in some aspect. More detailed information regarding the importance of using good performance indicators for evaluation may be found in the literature [5, 15, 27].

Here we use a performance indicator, denoted $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F} \to \mathcal{S}}$, that uses knowledge of the Pareto–optimal front [5]. Now, it can be shown that a single–objective indicator is not able to capture properly whether one approximation set is truly better than another [27]. However, because here the Pareto–optimal front is used to compare the approximation set with instead of another approximation set, optimality can be defined well using a single–objective indicator. This makes the $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F} \to \mathcal{S}}$ performance indicator a good, valid indicator [5]. The distance $d(\boldsymbol{x}^0, \boldsymbol{x}^1)$ between $\boldsymbol{x}^0$ and $\boldsymbol{x}^1$ is the Euclidean distance between the objective values $f(\boldsymbol{x}^0)$ and $f(\boldsymbol{x}^1)$. The $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F} \to \mathcal{S}}$ indicator computes the average of the distance to the closest solution in an approximation set $\mathcal{S}$ over all solutions in the Pareto–optimal set $\boldsymbol{\mathcal{P}_S}$. A smaller value for this performance indicator is preferable and a value of 0 is obtained if and only if the approximation set and the Pareto–optimal front are identical. This indicator describes how well the Pareto–optimal front is covered and thereby represents an intuitive trade–off between diversity and proximity (i.e. closeness to the Pareto–optimal front). Even if all points in the approximation set are on the Pareto–optimal front the indicator is not minimized unless the solutions in the approximation set are spread out perfectly.

Because the Pareto–optimal front may be continuous, a line integration over the Pareto front is required in the definition of the performance indicator. In a practical setting, it is easier to compute a uniformly sampled set of many solutions along the Pareto–optimal front and to use this discretized representation of $\boldsymbol{\mathcal{P}_F}$ instead. We have used this approach with 5000 uniformly sampled points:

$$D_{\mathcal{P}_F \to \mathcal{S}}(\mathcal{S}) = \frac{1}{|\mathcal{P}_S|} \sum_{x^1 \in \mathcal{P}_S} \min_{x^0 \in \mathcal{S}} \{d(x^0, x^1)\} \qquad (3)$$

5.4 Results

Figure 3 shows convergence graphs of the $D_{\mathcal{P}_F \to \mathcal{S}}$ indicator for all problems and all MOEDAs with a population size of 500 in different subpopulation configurations. A maximum of 10^6 evaluations was used where one evaluation involves evaluating both objectives. In this paper, we do not determine the minimally required resources (i.e. number of clusters and minimally required subpopulation size) because we focus on analyzing convergence properties.

On the relatively simple EC_i problems MIDEA alone outperforms its SDR–AVS and AVS counterparts if the subpopulation sizes are large enough. The reason for this is that if the subpopulations are large enough, variance scaling is not needed and hence only slows down convergence. Still, the addition of AVS allows the EDA to solve more problems reliably. Convergence on the BD_i problems is much better with AVS. If in addition SDR is used, faster convergence is obtained than when AVS is used alone for almost all problems and all subpopulation configurations.

Success rates can be determined as the percentage of times a MOEDA was able to reach a certain $D_{\mathcal{P}_F \to \mathcal{S}}$ value. In Table 2 all success rates are presented for $D_{\mathcal{P}_F \to \mathcal{S}} \leq 0.01$. For the problems in our test–suite, given the ranges of the objectives for the Pareto–optimal front configurations, a value of 0.01 for the $D_{\mathcal{P}_F \to \mathcal{S}}$ indicator corresponds to fronts that are quite close to the Pareto–optimal front. Examples of fronts with a $D_{\mathcal{P}_F \to \mathcal{S}}$ value of 0.01 are shown in Figure 4.

Table 2 confirms that the use of AVS results in better convergence. With 10 clusters of size 50, AVS–MIDEA obtains a success rate of 100%. Using multiple clusters indeed substantially helps to obtain better results. Moreover, smaller subpopulation sizes can be used with AVS. The required increase in variance to converge to the optimum no longer needs to come from a larger set of solutions to estimate the distribution from because the variance is artificially kept larger. This is in agreement with earlier, single–objective, results [11]. The only reason why AVS–MIDEA fails to reach the required $D_{\mathcal{P}_F \to \mathcal{S}}$ value for less clusters and a larger subpopulation size is that convergence is slower with AVS. However, without AVS, not all problems can be solved as can be seen from the results on problem BD_2.

MOEDA	Clusters	BD_1	BD_2	EC_1	EC_2	EC_3
MIDEA	2×250	0	0	100	100	100
AVS–MIDEA	2×250	100	0	0	14	0
SDR–AVS–MIDEA	2×250	100	0	100	69	0
MIDEA	5×100	33	0	100	100	0
AVS–MIDEA	5×100	100	100	0	28	28
SDR–AVS–MIDEA	5×100	100	100	100	100	100
MIDEA	10×50	0	0	0	0	0
AVS–MIDEA	10×50	100	100	100	100	100
SDR–AVS–MIDEA	10×50	100	100	100	100	76

Table 2: Success rates, i.e. the percentage of times a MIDEA variants obtained $D_{\mathcal{P}_F \to \mathcal{S}}$ indicator ≤ 0.01.

From Table 2 it can be seen that the addition of SDR improves the results of AVS. All problems can be solved reliably within the limit of the number of allowed evaluations, even for a configuration of larger subpopulation sizes of 100 divided over 5 subpopulations due to faster, but still reliable, convergence as can be seen from Figure 3. The results become slightly worse however if the subpopulation size becomes smaller (i.e. for 10 clusters of size 50 problem EC_3 cannot be solved in all 100 runs). Variance scaling is required when the subpopulation size becomes this small. As SDR reduces variance scaling the minimally required population size is slightly larger than when AVS is used without SDR. If the population size is large enough however, more efficient convergence is obtained by SDR–AVS–MIDEA than by AVS–MIDEA, while still being able to solve all problems, something that MIDEA alone, i.e. without AVS, cannot do.

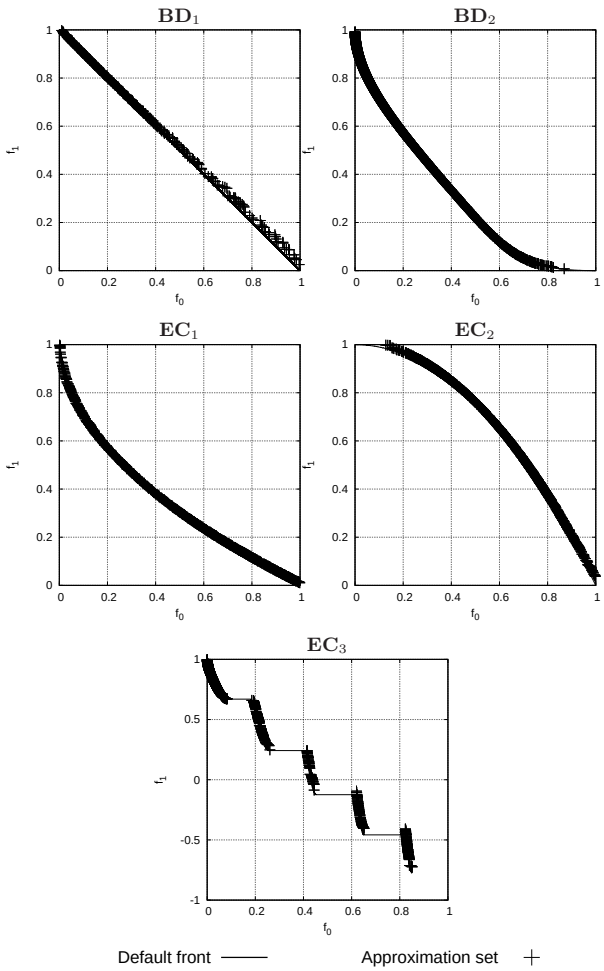

Figure 4: For all problems: the default front and an approximation set with a $D_{\mathcal{P}_F \to \mathcal{S}}$ indicator value of 0.01 obtained by SDR–AVS–MIDEA with a population size of 500 (5 clusters of size 100).

We finally note that the convergence results on BD_1 suggest that this problem can be solved without AVS although BD_1 contains Rosenbrock's function, which cannot be solved without AVS in the single–objective case. The reason for this lies in the difference between multi– and single–objective optimization. Because multiple non–dominated solutions are maintained along the front, the variance of the normal distributions is automatically larger. This increased variance allows the EDA to converge, albeit slowly, to the Pareto–optimal front. For this problem, this variance–increasing side–effect helps in finding the optimal front. This is however not always the case as was already hinted at in the introduction. The convergence behavior on problem BD_2 shows that use of multiple clusters does not increase the variance in the right direction and optimization fails.

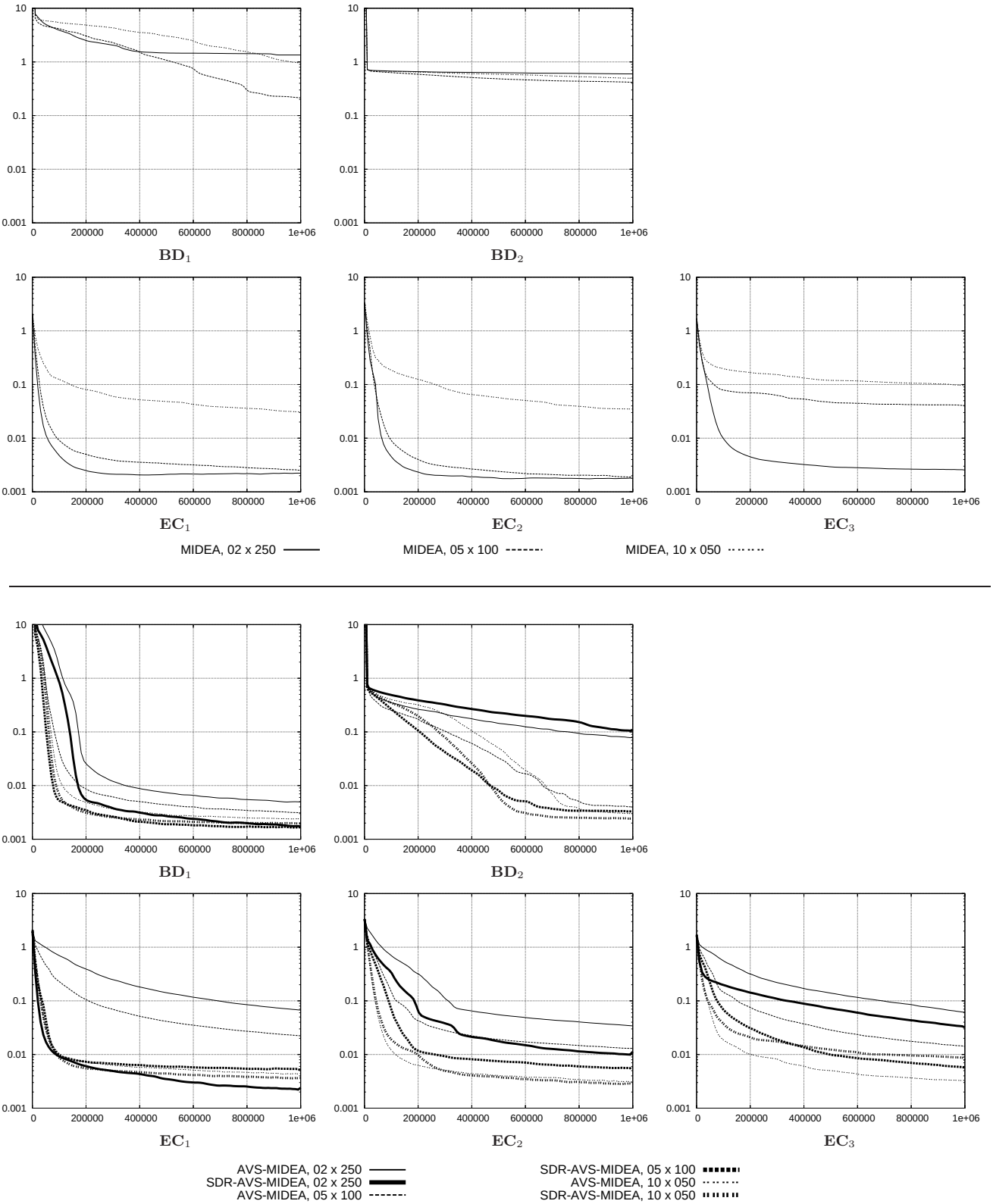

Figure 3: Convergence of all $\mathbb{M}$IDE$\mathbb{A}$ variants in various subpopulation configurations ($k \times n^{subpop}$) on all problems. Horizontal axis: number of evaluations (both objectives per evaluation). Vertical axis: $D_{\mathcal{P}_F \to \mathcal{S}}$.

6. CONCLUSIONS

We have investigated a principled approach to enlarging the class of problems that continuous MOEDAs can solve reliably. This approach is the use of adaptive variance scaling (AVS). In AVS, the variance of the estimated probability distribution can be enlarged beyond its maximum–likelihood estimate to prevent premature convergence. Because it is known that using spatially separated clusters helps multi–objective optimization, we have proposed a means to maintain spatially separated clusters throughout a run to be able to assign a separate AVS mechanism to each cluster.

Although more problems can be solved reliably with the addition of AVS, the overall convergence speed is reduced. We added a trigger, called standard–deviation ratio (SDR) that discriminates between improvements made close to the mean and improvements made far away from the mean. If improvements are made close to the mean, the variance is not scaled up any further. This trigger improves convergence speed while still allowing all problems to be solved.

Our results indicate that the addition of AVS and SDR to the $\mathbb{MIDEA}$ result in improved optimization behavior. From our results we cannot yet formulate guidelines for setting the parameters of the SDR–AVS–$\mathbb{MIDEA}$. A scalability study that reveals the minimally required number of clusters and subpopulation sizes is therefore important future research.

Overall, we conclude that adaptive variance scaling is an important and useful tool for designing more efficient multi–objective estimation–of–distribution algorithms.

7. REFERENCES

[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. Dover Publications, New York, New York, 1972.

[2] Th. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[3] P. A. N. Bosman and E. D. de Jong. Combining gradient techniques for numerical multi-objective evolutionary optimization. In W. B. Langdon et al., editors, *Proc. of the Genetic and Evolutionary Comp. Conf. – GECCO–2006*, pages 627–634, New York, New York, 2006. ACM.

[4] P. A. N. Bosman and D. Thierens. Advancing continuous IDEAs with mixture distributions and factorization selection metrics. In M. Pelikan and K. Sastry, editors, *Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Comp. Conf. GECCO–2001*, pages 208–212, San Francisco, California, 2001. Morgan Kaufmann.

[5] P. A. N. Bosman and D. Thierens. The balance between proximity and diversity in multi–objective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7:174–188, 2003.

[6] P. A. N. Bosman and D. Thierens. Multi–objective optimization with the naive MIDEA. In J. A. Lozano et al., editors, *Towards a New Evolutionary Computation. Advances in Estimation of Distribution Algorithms.*, pages 123–157. Springer, Berlin, 2006.

[7] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230, 1999.

[8] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer et al., editors, *Par. Prob. Solving from Nature – PPSN VI*, pages 849–858. Springer, 2000.

[9] C. M. Fonseca and P. J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evol. Computation*, 3(1):1–16, 1995.

[10] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison–Wesley, Reading, 1989.

[11] J. Grahl, P. A. N. Bosman, and F. Rothlauf. The correlation–triggered adaptive variance scaling idea. In M. Keijzer et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO–2006*, pages 397–404, New York, New York, 2006. ACM Press.

[12] J. Grahl, S. Minner, and F. Rothlauf. Behaviour of UMDAc with truncation selection on monotonous functions. In *Proc. of the Congress on Evol. Comp. – CEC–2005*, pages 2553–2559, Piscataway, New Jersey, 2005. IEEE Press.

[13] N. Hansen, S. D. Mller, and P. Koumoutsakos. Reducing the time complexiy of the derandomized evolution strategy with covariance matrix adaption. *Evolutionary Computation*, 11(1):1–18, 2003.

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

[15] J. Knowles and D. Corne. On metrics for comparing non–dominated sets. In *Proceedings of the 2002 Congress on Evol. Computation – CEC–2002*, pages 666–674, Piscataway, New Jersey, 2002. IEEE Press.

[16] P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation.* Kluwer Academic, London, 2001.

[17] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evol. Comp.*, 10(3):263–282, 2002.

[18] S. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

[19] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea. *Towards a New Evolutionary Computation. Advances in Estimation of Distribution Algorithms.* Springer–Verlag, Berlin, 2006.

[20] H. Mühlenbein and T. Mahnig. FDA – a scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evol. Comp.*, 7:353–376, 1999.

[21] J. Ocenasek, S. Kern, N. Hansen, and P. Koumoutsakos. A mixed bayesian optimization algorithm with variance adaptation. In X. Yao et al., editors, *Parallel Problem Solving from Nature – PPSN VIII*, pages 352–361, Berlin, 2004. Springer–Verlag.

[22] M. Pelikan and D. E. Goldberg. Genetic algorithms, clustering, and the breaking of symmetry. In M. Schoenauer et al., editors, *Par. Prob. Solving from Nature – PPSN VI*, pages 385–394. Springer, 2000.

[23] M. Pelikan, K. Sastry, and E. Cantú-Paz. *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications*. Springer, Berlin, 2006.

[24] D. Thierens. Scalability problems of simple genetic algorithms. *Evol. Computation*, 7(4):331–352, 1999.

[25] D. Thierens and P. A. N. Bosman. Multi–objective mixture–based iterated density estimation evolutionary algorithms. In L. Spector et al., editors, *Proc. of the Genetic and Evolutionary Computation Conference – GECCO–2001*, pages 663–670, San Francisco, California, 2001. Morgan Kaufmann.

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

[27] E. Zitzler, M. Laumanns, L. Thiele, C. M. Fonseca, and V. Grunert da Fonseca. Why quality assessment of multiobjective optimizers is difficult. In W. B. Langdon et al., editors, *Proc. of the Genetic and Evolutionary Computation Conference – GECCO–2002*, pages 666–674, San Francisco, California, 2002. Morgan Kaufmann.