The Anticipated Mean Shift and Cluster Registration in Mixture-based EDAs for Multi-Objective Optimization

Peter A.N. Bosman
Centrum Wiskunde & Informatica (CWI)
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Peter.Bosman@cwi.nl

ABSTRACT

It is known that in real-valued Single-Objective (SO) optimization with Gaussian Estimation-of-Distribution Algorithms (EDAs), it is important to take into account how distribution parameters change in subsequent generations to prevent inefficient convergence as a result of overfitting, especially if dependencies are modelled. We illustrate that in Multi-Objective (MO) optimization the risk of overfitting is even larger and only further increased if clustered variation is used, a technique often employed in Multi-Objective EDAs (MOEDAs) in the form of mixture modelling via clustering selected solutions in objective space. We point out that a technique previously used in EDAs to remove the risk of overfitting for SO optimization, the anticipated mean shift (AMS), can also be used in MO optimization if clusters in subsequent generations are registered. We propose to compute this registration explicitly. Although computationally more intensive than existing approaches, the effectiveness of AMS is thereby increased. We further propose a new clustering technique to improve mixture modelling in EDAs by 1) allowing clusters to overlap substantially and 2) assigning each cluster the same number of solutions. This allows any existing EDA to be transformed into a mixture-based version straightforwardly. Finally, we point out the benefit of injecting solutions obtained from running equal-capacity SO optimizers in synchronous parallel and investigate experimentally, using 9 well-known benchmark problems, the advantages of each of the techniques.

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

Estimation of Distribution Algorithms, Multi-Objective Optimization, Mixture Distribution, Anticipation

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.

1. INTRODUCTION

EDAs aim to exploit features of a problem's structure in a principled manner via probabilistic modelling. It is often assumed that a higher capacity of the distribution class used in an EDA automatically allows for a larger, and more complex, class of optimization problems to be solved efficiently. Merely enlarging this capacity, e.g. by allowing more dependencies to be modelled, isn't necessarily enough however. When using Gaussian (i.e. normal) distributions with maximum-likelihood estimates for instance, it is known that modelling dependencies may actually lead to overfitting the selected solutions, which, in turn, results in an inefficient alignment of the distribution with the direction of improvement in the problem landscape [5, 6, 7, 8]. For this reason, the direction in which the distribution has shifted in subsequent generations must be considered. This is done using adaptive mechanisms that span multiple generations such as the Anticipated Mean Shift (AMS) [1] approach in EDAs and the evolution path and the estimation of covariances using the mean in the previous generation in CMA-ES [8].

For single-objective (SO) optimization, these algorithms are highly efficient. Many optimization problems in practice however are multi-objective (MO). In MO optimization, the optimum is no longer a single solution but a set of solutions, called the optimal Pareto front. This is because many solutions may be equally good, e.g. solution $\boldsymbol{a}$ may be better in the first objective than solution $\boldsymbol{b}$, but worse in the second objective. Population-based methods such as evolutionary algorithms (EAs) are commonly accepted to be well-suited for solving MO problems [4]. Because a set of solutions is used, EAs can spread their search bias along the Pareto front and thereby prevent many re-computations that are involved if a single point on the Pareto front is repeatedly targeted using an approach that only considers a single solution.

Considering EDAs, mixture distributions are of particular interest when solving MO optimization problems because they can spread the search intensity along the Pareto front, allowing more focused exploitation of problem structure in different regions of the objective space [2, 12]. To obtain high-quality solutions, exploiting dependencies in each region may be necessary, but the configuration of these dependencies or the values for the problem variables may be very different in each region. Probabilistic dependency modelling may be less effective if it is the same in each region.

It is the focus of this paper to study more closely the relation between spreading the search distribution in EDAs using mixture distributions and the observed pressure towards finding better (i.e. Pareto-dominating) solutions. In MO op-

timization, the number of equally-preferable solutions can easily be larger than the population size, causing the variance of the estimated distribution to quickly become focused on the variety *within* sets of equally-preferable solutions instead of the variety *between* such sets, i.e. the direction of improvement in the MO fitness landscape. Non-zero variance along such directions is required for any EDA to have a substantial probability of sampling better solutions. If higher-order dependencies can be modelled, the risk of fitting only solutions of equal preference becomes only larger because a more accurate probabilistic representation of the selected solutions is possible, especially in the case of real-valued objectives because then there may be an infinite number of equally-preferable solutions. Arguably, premature convergence and inefficient performance are then much more likely, making this an important topic to study more closely.

2. MULTI-OBJECTIVE OPTIMIZATION

We assume to have m objective functions $f_i(\boldsymbol{x})$, $i \in \{0, 1, \ldots, m-1\}$ and, without loss of generality, we assume that the goal is to *minimize* all objectives.

A solution $\boldsymbol{x}^0$ is said to (Pareto) *dominate* a solution $\boldsymbol{x}^1$ (denoted $\boldsymbol{x}^0 \succ \boldsymbol{x}^1$) if and only if $f_i(\boldsymbol{x}^0) \leq f_i(\boldsymbol{x}^1)$ holds for all $i \in \{0, 1, \ldots, m-1\}$ and $f_i(\boldsymbol{x}^0) < f_i(\boldsymbol{x}^1)$ holds for at least one $i \in \{0, 1, \ldots, m-1\}$. A *Pareto set* of size n then is a set of solutions $\boldsymbol{x}^j$, $j \in \{0, 1, \ldots, n-1\}$ for which no solution dominates any other solution, i.e. there are no $j, k \in \{0, 1, \ldots, n-1\}$ such that $\boldsymbol{x}^j \succ \boldsymbol{x}^k$ holds. A *Pareto front* corresponding to a Pareto set is the set of all m-dimensional objective function values corresponding to the solutions, i.e. the set of all $\boldsymbol{f}(\boldsymbol{x}^j)$, $j \in \{0, 1, \ldots, n-1\}$.

A solution $\boldsymbol{x}^0$ is said to be *Pareto optimal* if and only if there is no other $\boldsymbol{x}^1$ such that $\boldsymbol{x}^1 \succ \boldsymbol{x}^0$ holds. Further, the *optimal Pareto set* is the set of all Pareto-optimal solutions and the *optimal Pareto front* is the Pareto front that corresponds to the optimal Pareto set. We denote the optimal Pareto set by $\boldsymbol{\mathcal{P}_S}$ and the optimal Pareto front by $\boldsymbol{\mathcal{P}_F}$.

3. CLUSTERED VARIATION

Instead of using one population, multiple populations can be used. With the exception of selection, a completely separate EA is run for each subpopulation. Because selection is performed on all solutions in all populations the generations are synchronized and the populations can also be thought of as subpopulations. This approach is taken in SDR-AVS-$\mathbb{M}$IDEA [3] and in MO-CMA-ES [11].

Not all populations necessarily then get the same number of selected solutions. For some populations, none of the generated solutions may even be selected in the next generation. In that case, the population will have to be reset somehow, for instance by copying solutions from other populations. Also, all adaptive mechanisms that span multiple generations will have to be reset for the disappearing population. This is the case for SDR-AVS-$\mathbb{M}$IDEA [3]. One way to overcome this problem is to restrict the population size to be of size 1. This is the case for in MO-CMA-ES [11] where a $(1, 1)$ strategy is used. This restriction however doesn't allow other existing SO population-based methods to be extended to the MO case in a straightforward manner.

Clustered variation can also be performed using only one population. The selected solutions are then first clustered. Subsequently, the actual variation takes place by considering only individuals in the same cluster, i.e. a mating restriction is employed. To ensure that the spatial separation of the search bias is obtained in the objective space, clustering should be performed on the basis of objective values. In EDAs, this corresponds to using a mixture distribution. A mixture probability distribution is a weighted sum of k probability distributions. Let $\boldsymbol{X}$ be the random variable that represents the parameter space of the problem at hand. A mixture probability distribution is then defined by $\sum_{i=0}^{k-1} \beta_i P^i(\boldsymbol{X})$, $\beta_i > 0$, $i \in \{0, \ldots, k-1\}$ 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.

Using mixture probability distributions instead of subpopulations is probabilistically a superior approach because all data is used each generation to compute the distribution. Obtaining mixture distributions by clustering the selected solutions and estimating a probability distribution in each cluster separately is an approach taken in various MOEDAs, e.g. in $\mathbb{M}$IDEA [2] and in mohBOA [12]. The main difference between these two approaches, besides employing Gaussians for real-valued solutions versus employing decision graphs in Bayesian factorizations for discrete solutions, is that in $\mathbb{M}$IDEA a different clustering algorithm is used (leader clustering) than in mohBOA (k-means). We refer the interested reader for details on these clustering algorithms to the respective literature. Although asymptotically these clustering algorithms have the same computational complexity, the k-means clustering algorithm loops over the data more than once, requiring more time, but typically resulting in a superior clustering result with less variation in cluster sizes.

The clustering methods used so far do not necessarily result in a clustering where each cluster has equal size. This can give similar problems as when using multiple populations. For a straightforward extension of existing EAs, it is convenient to know for sure that cluster sizes are uniform and what this size is. To this end, we propose the following mix of the leader and the k-means clustering algorithms.

First, a nearest-neighbour heuristic is used to select k leaders that are spread as well as possible: the first leader is chosen as a solution with a maximum value for a randomly chosen objective. For all remaining solutions, the nearest-neighbor distance is computed to the single leader and the one with the largest distance is chosen as the next leader. The distances for the remaining solutions are updated by checking whether the distance to the new leader is smaller than the currently stored nearest-neighbour distance. These last two steps are repeated until k leaders are selected. Second, these solutions serve as the initial cluster means for k-means clustering. Third, the distance from each selected solution to the final cluster means is computed. After sorting then, for each cluster the closest c solutions are finally assigned to that cluster, ensuring that each cluster consists of exactly c solutions. Because sorting and the final assignment is done independently for each cluster, some solutions may be assigned to multiple clusters whereas other solutions are not assigned at all. The probability of this happening can be reduced by forcing the clusters to overlap by setting $c > \frac{1}{k}|\boldsymbol{\mathcal{S}}|$ where $\boldsymbol{\mathcal{S}}$ is the set of selected solutions. Specifically, we propose to use $c = \frac{2}{k}|\boldsymbol{\mathcal{S}}|$, resulting in substantial expected overlap between neighboring clusters. This increases the expected density in the usual void between the boundaries of clusters in the objective space, thereby increasing the probability of finding a good, uniform spread of solu-

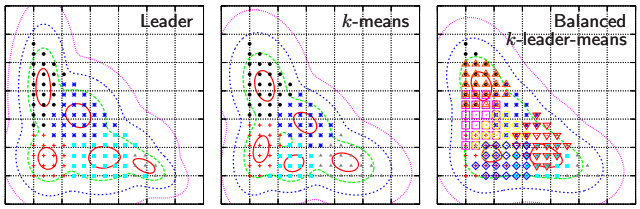

Figure 1: Three clustering algorithms and density contours of the associated Gaussian mixture.

tions faster. Further, twice the number of clusters can be used in this way, given the same population size.

Figure 1 shows results of clustering 105 samples in a triangle, reminiscent of a selection result on a $2D$ slope, i.e. minimizing $x_0 + x_1$. Also shown are the density contours of the associated Gaussian mixture that is obtained by estimating a Gaussian distribution in each cluster. For the leader and the k-means clustering algorithm, 5 clusters are computed. For the proposed balanced k-leader-means (BKLM) clustering algorithm, 10 clusters are computed. An increase in uniformity of the density estimate can be observed with an increase in clustering effort, with the smoothest density estimate obtained using BKLM. The problem of unequal cluster sizes also diminishes with increased clustering effort. The most uneven result was found for leader clustering: 29, 27, 23, 19 and 8, followed by k-means: 27, 22, 22, 20, 15 and finally BKLM with all equal cluster sizes of 21.

Finally, we remark that clustering in MOEDAs should compute distances based on *normalized* objective values to remove the influence of differently scaled objectives. To this end, first the minimum $f_i^{\min}$ and maximum $f_i^{\max}$ values for each objective i can be computed from all selected solutions. A point in objective space $f(\boldsymbol{x})$ then can be scaled linearly to the observed ranges, i.e. $(f(\boldsymbol{x}) - f_i^{\min})/(f_i^{\max} - f_i^{\min})$.

4. CLUSTER REGISTRATION

An important part of state-of-the-art variation operators are adaptive mechanisms that span multiple generations such as the Anticipated Mean Shift (AMS) [1] approach in EDAs and the evolution path and the estimation of the covariance matrix based on the mean in the previous generation in CMA-ES [8]. The contribution of these mechanisms strongly depends on a correlation to exist between the sets of solutions in subsequent generations from which the models are built. By re-applying clustering each generation however, in principle there is no spatial relation between clusters in subsequent generations. Even if the clustering algorithm has low variation when applied twice to the same data, the final enumeration of the clusters does not guarantee at all that cluster i in generation $t-1$ is near cluster i in generation t. Therefore, some form of registration is required that determines the best correspondence between clusters in subsequent generations. An implicit form of registration is achieved by assigning each newly generated solution to the cluster to which it is nearest (i.e. the highest density). This approach is taken in SDR-AVS-MIDEA [3]. Once new solutions cannot be assigned to a particular cluster anymore because it has become too large already (i.e. larger than the predefined subpopulation size), suboptimal cluster assignments can be made. Over multiple generations, the spatial separation can then degrade, resulting in clusters moving across the Pareto front as can for instance be observed in Figure 2. This problem can not be overcome by using a

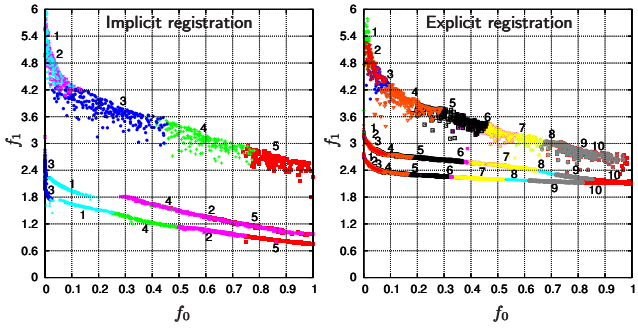

Figure 2: Clustering of selected solutions in different generations using implicit and explicit registration of 5 and 10 clusters respectively and estimating a Gaussian with a full covariance matrix per cluster.

population size of 1 as in MO-CMA-ES [11] unless an explicit registration is performed. Therefore, we propose to explicitly compute a registration between clusters in subsequent generations. The approach to this end that we propose here is not specific for Gaussian EDAs and can therefore be applied to any clustered or multi-population algorithm.

The goal of explicit cluster registration is to re-assign the cluster indices of the current generation t such that cluster i in generation t is the cluster that is closest to cluster i in the previous generation $t-1$. To this end, we propose an algorithm that first computes all distances between clusters in generation t and generation $t-1$. The distance between two clusters is taken to be the smallest distance between any solution in the one cluster and any solution in the other cluster. Also all cluster distances are computed between the clusters in generation t and between the clusters in generation $t-1$. Then, the algorithm repeatedly selects $r \leq k$ clusters to be registered, that is, r clusters in generation t and r clusters in generation $t-1$. To this end, first the two still-unregistered clusters in generation t are determined that are the farthest apart. One of these two far-apart clusters is randomly selected as well as the still-unregistered cluster in generation $t-1$ that is closest to it. The $r-1$ nearest neighbours of these clusters are then determined in the set of still-unregistered clusters of their respective generations, leading to two subsets of r clusters to be registered. To register subsets of clusters, all possible $r!$ permutations for the set of clusters in generation t are considered and the permutation is selected for which the sum of the distances between the matched clusters is minimal. Subset registration is then repeated until all clusters are registered.

The reason for using subset registration with $r \leq k$ instead of $r = k$ is that subset registration is performed by enumerating permutations. As this number grows factorially fast, exact optimization via enumeration of all possible permutations can only be done for small values of r. Still, we found that r can be set large enough (we used $r = 10$) to substantially reduce the risk of suboptimal registration without requiring more time than other parts in model-building.

Figure 2 shows clusters in different generations (1, 30 and 60) using the same number of solutions for implicit registration and explicit registration on the well-known benchmark problem EC_1. For implicit registration, $k = 5$ subpopulations are used. For explicit registration, BKLM is used with $k = 10$ clusters. In each cluster a Gaussian distribution is estimated using a full covariance matrix without further adaptive enhancements. The superiorly smooth front and

stable registration over many generations is clear for explicit registration, but so is the lack of front progress as a result of overfitting the selected solutions with more involved mixture estimates. Next, we specifically target this issue.

5. GAUSSIANS, AMS, SDR, AVS AND MO

Estimating a Gaussian distribution only using the selected solutions of the current generation, the density contours can become aligned with directions in which only solutions of similar quality can be found. Methods that only adaptively scale the covariance matrix, such as SDR-AVS, do not help much as they almost solely increases search effort in the futile direction perpendicular to the direction of improvement. In SDR-AVS, a distribution multiplier $c^{\text{Multiplier}}$ is maintained by which the covariance matrix is multiplied each generation. This multiplier is scaled up if improvements are found that are more than standard-deviation away from the mean and scaled down if no improvements are found (for more details, see [1]). This misalignment behavior is already known to occur in SO optimization with EDAs [1, 8], but the same issue can occur in MO optimization because it is a direct consequence of selecting solutions of similar quality, regardless of the number of objectives.

This inefficient behavior is illustrated in Figure 3 on a two-dimensional and two-objective minimization problem defined by $f_0(\boldsymbol{x}) = \frac{1}{2}(x_0^2 + (x_1 - 1.0)^2)$ and $f_1(\boldsymbol{x}) = \frac{1}{2}((x_0 - 1.0)^2 + x_1^2)$. The optimal Pareto front is convex and defined by $x_0 = 1 - x_1$ and $f_1 = f_0 - 2\sqrt{f_0} + 1$. By initializing the population in the initialization range (IR) $[0.9; 1.0]^2$, the better solutions form a rotated V-shape in the lower-left triangle of the IR. Using a maximum-likelihood estimate, the distribution thereby becomes misaligned with the direction of improvement. Although the variance is adaptively scaled up, the misalignment prevents the MOEDA from efficiently locating solutions closer to the optimal Pareto front.

One way to overcome this problem, is to use the Anticipated Mean Shift (AMS) [1]. The AMS is computed as the difference between the means of subsequent generations, i.e. $\hat{\boldsymbol{\mu}}^{\text{Shift}}(t) = \hat{\boldsymbol{\mu}}(t) - \hat{\boldsymbol{\mu}}(t-1)$. A part, specifically $\alpha 100\%$, of the newly sampled solutions is then moved in the direction of the AMS: $\boldsymbol{x} \leftarrow \boldsymbol{x} + 2\hat{\boldsymbol{\mu}}^{\text{Shift}}(t)$. The rationale is that solutions changed by AMS are further down the slope. Selecting those solutions as well as solutions not changed by AMS aligns the distribution estimate better with the direction of improvement. In a population of size n where $\lfloor \tau n \rfloor$ solutions are selected, n^{elitist} solutions are maintained and $n - n^{\text{elitist}}$ new solutions are generated, proportioning the selected solutions perfectly between unaltered and AMS-altered solutions requires $\alpha(n - n^{\text{elitist}}) = \frac{1}{2}\tau n$ and thus $\alpha = \frac{1}{2}\tau \frac{n}{n - n^{\text{elitist}}}$. A combination of AMS with SDR and AVS has been termed AMaLGaM (Adapted Maximum-Likelihood Gaussian Model) [1] in which traversing a slope is further sped up by multiplying the movement of solutions in the direction of the AMS by the same multiplier used for the covariance matrix, i.e. $\boldsymbol{x} \leftarrow \boldsymbol{x} + c^{\text{Multiplier}} 2\hat{\boldsymbol{\mu}}^{\text{Shift}}(t)$.

The effect of adding AMS, i.e. using AMaLGaM, is shown for the example problem in Figure 3. In parameter space, the Gaussian is quickly adaptively re-aligned with the direction of Pareto-improvement. In objective space the variance towards the optimal Pareto front remains substantial, causing the density to already start spreading along the optimal Pareto front within the first 7 generations.

AMS, SDR and AVS can all be applied directly in combination with mixture distributions if a correspondence between the clusters in subsequent generations exists. For each cluster then, a separate AMS, SDR and AVS mechanism can be used. For the use of SDR-AVS without AMS however, the resulting performance of the MOEDA when going from a single distribution to a mixture distribution can become even worse. In Figure 3 it can clearly be seen that the problem of overfitting the selected solutions is even more problematic. In each cluster, the selected solutions can be fitted even more closely, resulting in an overall better fit, but less progress in terms of optimization. The addition of AMS for each cluster separately changes this behavior completely. Similar to using a single distribution, within a few generations the most important direction of improvement is detected and the density of the estimated distribution is re-aligned to efficiently find Pareto-improving solutions. The density in the objective space also shows that in the last few generations a more uniform density estimate is obtained only in the vicinity of the Pareto front, whereas for a single cluster the density in objective space is spread out even across large parts of the objective space that are inferior.

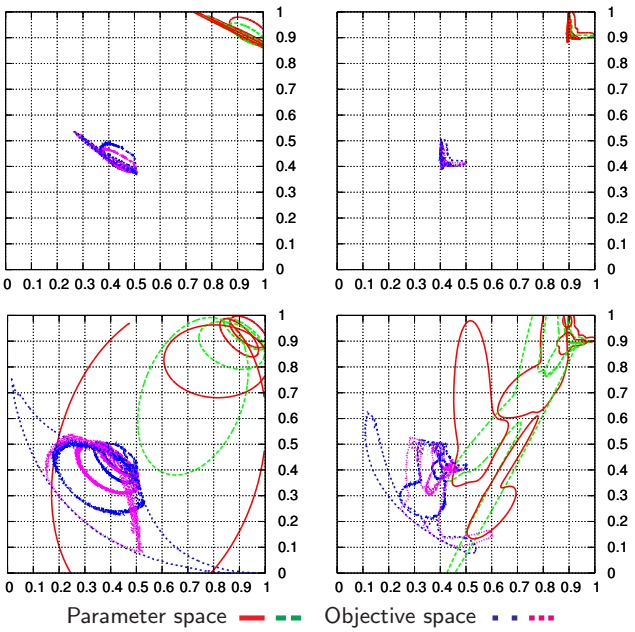

Figure 3: 95%-contours of the estimated distribution in the first 7 generations of typical MOEDA runs with either a single Gaussian (left column) or a mixture (right column) and with either SDR-AVS (top row) or AMaLGaM (bottom row), on the example problem with IR $[0.9; 1.0]^2$. Subsequent generations alternatingly use solid and dashed lines. The estimations are shown both in parameter space (red and green) and objective space (blue and pink).

6. MAMALGAM-X

We call the composition of the techniques proposed above, i.e. the MOEDA illustrated in the bottom-right in Figure 3, MAMaLGaM-X (Multi-objective AMaLGaM-miXture) and summarize its operational description below.

Given a population of size n, $\lfloor \tau n \rfloor$, $\tau \in [\frac{1}{n}, 1]$, solutions are selected and clustered using BKLM, giving cluster sizes of $\frac{2}{k}\lfloor \tau n \rfloor$. Selection is performed by computing domination ranks and then selecting the lowest ranks that fit within the

maximum of $\lfloor \tau n \rfloor$. From the rank that crosses this boundary, the same nearest neighbour heuristic is used to fill the selected set with as is used to select the k leaders in BKLM.

An elitist archive is maintained, storing all currently non-dominated solutions. Because the objectives are real-valued, there are typically infinitely many non-dominated solutions possible. To prevent the archive from growing to an extreme size, the objective space is discretized into hypercubes. Only one solution per hypercube is allowed in the archive. Newly generated solutions are compared to the solutions in the archive. If a new solution is dominated by any archive solution, it is not entered. If a new solution is not dominated, it is added to the archive if the hypercube that it resides in does not already contain a solution or if it dominates that particular solution. When a new solution is entered, all archive solutions that are dominated by it, are removed.

After clustering and subsequently performing explicit cluster registration, a Gaussian distribution is estimated in each of the clusters and adapted using the combination of AMS, SDR and AVS as in AMaLGaM [1] with two minor differences. 1) The AVS scheme is based upon whether improvements are found. After generating new solutions, each newly generated solution is re-associated with the cluster to which it is closest in objective space. An improvement is said to be obtained for cluster i if any new solution associated with cluster i is added to the archive. 2) The SDR scheme computes, for each cluster, the average of all improvements associated with that cluster and checks whether the average lies beyond one standard deviation. Because here improvements can be obtained in different regions, the ratio of the average improvement is less informative. Instead, we therefore compute the average ratio of the improvements.

Keeping elitist solutions in the population can contribute to improved convergence. Therefore, each solution in the elitist archive is associated with its nearest cluster. For each cluster, at most $\frac{1}{k}\lfloor \tau n \rfloor$ of its associated elitist solutions are copied to the population. If there are more elitist solutions, the same nearest-neighbour heuristic is used as in selection. Finally, each cluster generates equally many solutions, corresponding to uniform mixture coefficients $\beta_i = \frac{1}{k}$. Depending on how many elitist solutions were copied to the population, at least $n - \lfloor \tau n \rfloor$ new solutions are thereby generated.

7. SYNCHRONOUS PARALLEL SOEDAS

Although clustered variation spreads the search bias, MO selection still focuses exploitation on all objectives at the same time, reducing pressure towards finding Pareto improvements. It may therefore be beneficial to add expert search bias in the form of separate SO optimization of the m objectives. In SO optimization there are typically less problems with maintaining pressure on finding improvements.

A combination of MO optimization and SO optimization has been proposed before [10]. There, $m+1$ equal-sized populations are used. Here, we propose to set the population size for each of the m SO optimizers equal to the cluster size in the MO population. For MAMaLGaM-X this amounts to an overall population size of $n + \frac{2mn}{k}$. We further propose to use an EA for SO optimization that is similar to the MO optimizer being used, i.e. using the same variation operator and same selection intensity. In this way, given enough clusters, the rate of convergence in each cluster is expected to be similar, resulting in better-aligned support of the SO optimizers in terms of convergence. Furthermore, in [10] solutions are migrated from the SO populations to the MO population and vice versa. Assuming competent SO optimizers however, this may only reduce the effectiveness of the SO optimizers. We therefore propose to only add the best solutions found by SO optimizers in each generation to the archive of the MO optimizer. By injecting the best solutions found by the SO optimizers for the different objectives into the elitist archive the pressure of the SO optimizers to find improvements can filter through to the MO optimizer. Also, the search bias of the MO optimizer is spread out towards the edges of the Pareto front, i.e. where the SO optimizers are, ensuring that no unnecessary gap appears between solutions found by the SO optimizers and solutions found by the MO optimizer. In the remainder we will refer to the SO-extended version of MAMaLGaM-X by MAMaLGaM-X$^+$.

8. EXPERIMENTS

8.1 Test suite

The definitions of the problems in our multi-objective optimization problem test suite are presented in Table 1.

The first two problems we use are the easiest. They are generalizations of the MED (Multiple Euclidean Distances) problems [9]. Each objective is similarly scaled. There are furthermore no constraints and no local Pareto fronts, making the problem relatively simple, comparable to the sphere function in real-valued SO optimization. The initialization range (IR) of $[-1; 1]$ is not a constraint. The optimal Pareto front for GM_1 is convex; for GM_2 it is concave.

We also used the well-known problems[1] EC_i, $i \in \{1, 2, 3, 4, 6\}$. The IRs of the EC_i problems are also constraints. These problems differ from the GM problems in that the objectives are not similarly defined and not similarly scaled. For more details about these functions, see [13].

The final two problems come from more recent literature on real-valued MO optimization [3] and are labeled BD_i, $i \in \{1, 2\}$. 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. Function BD_2 is harder than BD_1 in that the objective functions overlap in all variables instead of only in x_0. Further, the IR of x_0 in function BD_1 is also a constraint. 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 $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F}\rightarrow\mathcal{S}}$ indicator (which is explained in the next Section) on all problems corresponds to a similar front quality on all problems.

To avoid artifacts resulting from boundary-repair methods, the sampling procedure in all MOEDAs is constructed such that solutions that are out of bounds are rejected.

8.2 Measuring performance

We consider the elitist archive upon termination to be the outcome of a MOEDA and refer to it as an approximation set, denoted $\mathcal{S}$. To measure performance the $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F}\rightarrow\mathcal{S}}$ performance indicator is computed. This performance indicator computes the average distance over all points in the optimal Pareto front $\boldsymbol{\mathcal{P}_F}$ to the nearest point in $\mathcal{S}$: $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F}\rightarrow\mathcal{S}}(\mathcal{S}) = \frac{1}{|\boldsymbol{\mathcal{P}_F}|}\sum_{\boldsymbol{f}^1\in\boldsymbol{\mathcal{P}_F}} \min_{\boldsymbol{f}^0\in\mathcal{S}}\{d(\boldsymbol{f}^0, \boldsymbol{f}^1)\}$ where $\boldsymbol{f}$ is a point in objective space and $d(\cdot,\cdot)$ computes Euclidean distance. A smaller $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F}\rightarrow\mathcal{S}}$ value is preferable and a value

[1]These problems are also known as ZDT_i.

Name	Objectives	IR
GM_1	$f_0 = \left\|\frac{1}{2}(\boldsymbol{x}-\boldsymbol{c}^0)\right\|^d, \quad f_1 = \left\|\frac{1}{2}(\boldsymbol{x}-\boldsymbol{c}^1)\right\|^d$ $\boldsymbol{c}^0 = (1,0,0,\ldots), \quad \boldsymbol{c}^1 = (0,1,0,0,\ldots), \quad d = 2$	$[-1;1]^{10}$ $(l = 10)$
GM_2	$f_0 = \left\|\frac{1}{2}(\boldsymbol{x}-\boldsymbol{c}^0)\right\|^d, \quad f_1 = \left\|\frac{1}{2}(\boldsymbol{x}-\boldsymbol{c}^1)\right\|^d$ $\boldsymbol{c}^0 = (1,0,0,\ldots), \quad \boldsymbol{c}^1 = (0,1,0,0,\ldots), \quad d = \frac{1}{2}$	$[-1;1]^{10}$ $(l = 10)$
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	$f_0 = x_0, \quad f_1 = \gamma\left(1-\sqrt{f_0/\gamma}-(f_0/\gamma)\sin(10\pi f_0)\right)$ $\gamma = 1 + 9\left(\sum_{i=1}^{l-1} x_i/(l-1)\right)$	$[0;1]^{30}$ $(l = 30)$
EC_4	$f_0 = x_0, \quad f_1 = \gamma\left(1-\sqrt{f_0/\gamma}\right)$ $\gamma = 1 + 10(l-1) + \sum_{i=1}^{l-1}\left(x_i^2 - 10\cos(4\pi x_i)\right)$	$[-1;1]\times$ $[-5;5]^9$ $(l = 10)$
EC_6	$f_0 = 1 - e^{-4x_0}\sin^6(6\pi x_0), \quad f_1 = \gamma\left(1-(f_0/\gamma)^2\right)$ $\gamma = 1 + 9\left(\sum_{i=1}^{l-1} x_i/(l-1)\right)^{0.25}$	$[0;1]^{10}$ $(l = 10)$
BD_1	$f_0 = x_0, \quad f_1 = 1 - x_0 + \gamma$ $\gamma = \sum_{i=1}^{l-2}\left(100(x_{i+1}-x_i^2)^2 + (1-x_i)^2\right)$	$[0;1]\times$ $[-5.12;5.12]^9$ $(l = 10)$
BD_2^s	$f_0 = \frac{1}{l}\sum_{i=0}^{l-1} x_i^2$ $f_1 = \frac{1}{l-1}\sum_{i=0}^{l-2}\left(100(x_{i+1}-x_i^2)^2 + (1-x_i)^2\right)$	$[-5.12;5.12]^{10}$ $(l = 10)$

Table 1: The MO problem test suite.

of 0 is obtained if and only if the approximation set and the optimal Pareto front are identical. This indicator is useful for evaluating performance if the optimum is known because it describes how well the optimal Pareto front is covered and thereby represents an intuitive trade-off between the diversity of $\mathcal{S}$ and its proximity (i.e. closeness to the optimal Pareto front). Even if all points in the $\mathcal{S}$ are on the optimal Pareto front the indicator is not minimized unless the solutions in the approximation set are spread out perfectly. Because the optimal Pareto front may be continuous, there are infinitely many solutions possible on the optimal Pareto front. Therefore, we computed 5000 uniformly sampled solutions along the optimal Pareto front to use as a discretized version of $\boldsymbol{\mathcal{P}_F}$ for a high-quality approximation.

For the problems in our test-suite, given the ranges of the objectives for the optimal Pareto front configurations, a value of 0.01 for the $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F}\to\mathcal{S}}$ indicator corresponds to fronts that are quite close to the optimal Pareto front. Fronts that have a $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F}\to\mathcal{S}}$ value of 0.01 can be seen in Figure 4.

8.3 Results

All presented results are averaged over 30 runs. The subpopulation or cluster sizes were set according to guidelines from recent literature on SO [1]. For the two different problem sizes in our test suite, i.e. $l = 10$ and $l = 30$, this boils down to cluster sizes of 112 and 510 respectively for the full-covariance matrix, 52 and 99 for the Bayesian factorization and 32 and 55 for the univariate factorization. Both MOEDAs were given the same overall population size, meaning that twice the number of clusters could be used in MAMaLGaM-X (see Section 3), i.e. the population size in MAMaLGaM-X variants is $\frac{1}{2}k$ times the cluster size whereas the population size in SDR-AVS-MIDEA variants is k times the cluster size. The discretization of the objectives into hypercubes for the elitist archive is set to 10^{-3}. We compared SDR-AVS-MIDEA using implicit cluster registration with MAMaLGaM-X and MAMaLGaM-X$^+$ using explicit cluster registration. We observe the average convergence of the $\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F}\to\mathcal{S}}$ metric to study the impact of the various techniques proposed in combination with estimating Gaussian distributions either modelling all dependencies (i.e. using a full covariance matrix), a subset of all dependencies via greedy Bayesian factorization learning (which is a common approach in EDAs, see, e.g. [1, 2, 12]), or modelling no dependencies at all (i.e. using the univariate factorization). In case of the Bayesian factorization, we limited the maximum number of parents per variable to 5.

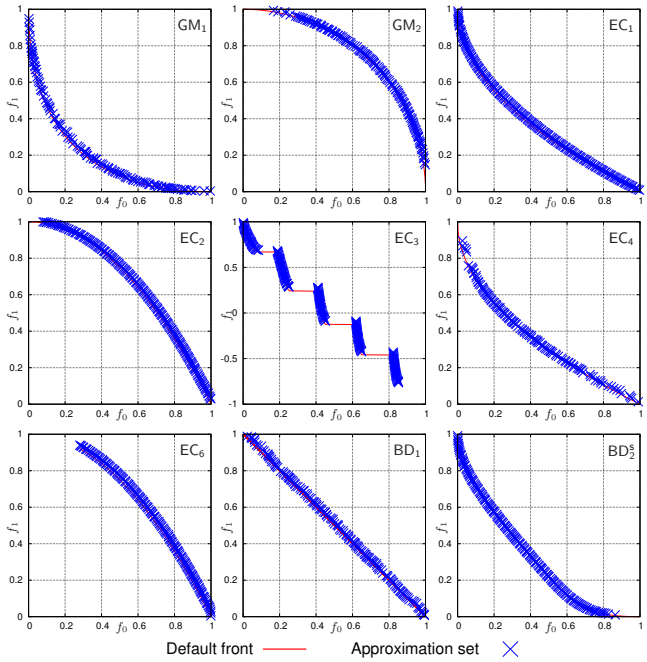

Figure 4: Default fronts and approximation sets obtained with MAMaLGaM-X$^+$ ($\boldsymbol{D}_{\boldsymbol{\mathcal{P}_F}\to\mathcal{S}} = 0.01, k = 20$).

In Figure 5 the convergence of successful runs of both MOEDAs is shown on EC_1 and EC_6 with and without the use of AMS. These results were found to be exemplary of the results on all problems. If not all runs were successful, the average convergence over all unsuccessful runs is also shown. A run is defined to be successful if a value of 0.01 was reached within the limit of 10^6 function evaluations. The convergence results without AMS are inferior. This is especially the case if the full covariance matrix is used, i.e. when overfitting is most likely. Overfitting is also more likely if BKLM clustering is used and consequently, without AMS, MAMaLGaM-X performs the worst. However, combined with explicit cluster registration, AMS has a tremendous impact on the performance of MAMaLGaM-X. AMS also speeds up SDR-AVS-MIDEA, albeit not as profoundly. AMS further can be seen to positively influence the convergence of both MOEDAs if only a subset of all possible dependencies is estimated. The impact is then smaller though because the density-misalignment problem associated with overfitting isn't there. Overall, MAMaLGaM-X (with AMS) has the best convergence behavior for all variants of dependency processing due to the use of the BKLM method combined with explicit registration.

A similar positive influence by AMS was observed on all problems, for which reason we refrain from presenting further convergence graphs for results obtained without AMS. Instead, results are summarized using success rates (within the limit of 10^6 evaluations) and presented in Table 2. These results confirm that using AMS results in better perfor-

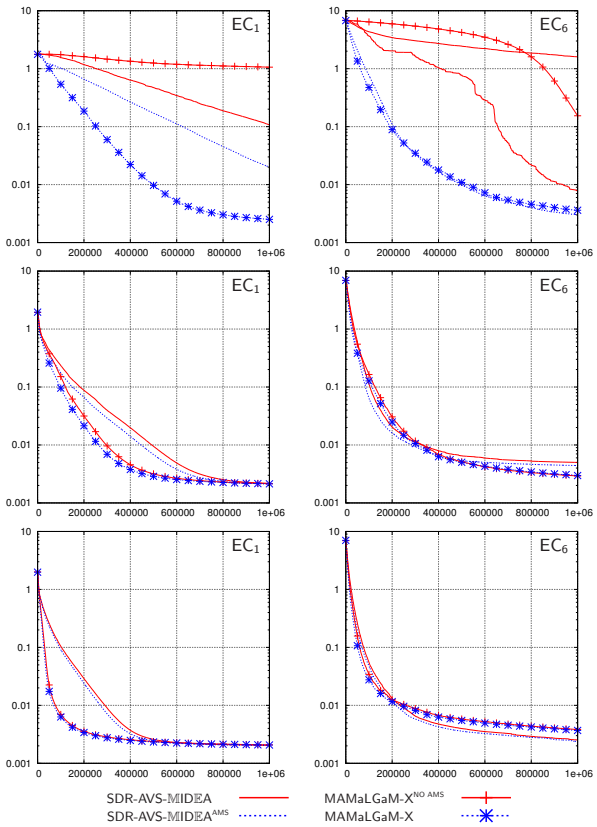

Figure 5: Average performance of SDR-AVS-MIDEA ($k = 10$) and MAMaLGaM-X ($k = 20$) with and without AMS, on two problems, estimating full covariance matrices (top row), Bayesian factorizations (center row) and no covariances (bottom row). Horizontal axis: number of evaluations (both objectives per evaluation). Vertical axis: $D_{\mathcal{P}_F \to \mathcal{S}}$. For each algorithm averages are shown both for succesful runs and unsuccesful runs, giving double occurrences of lines if some runs were unsuccesful.

mance. The table also shows the severity of the impact of overfitting. Going from low-order dependency learning to learning the full covariance matrix, one would expect only higher success rates. Without the use of AMS however, the success rates almost always drop, often to near 0% success. With AMS however, the overfitting problem is relieved and the intuition of being able to solve a larger class of problems reliably by estimating dependencies again can be seen, obtaining high success rates (given enough clusters and explicit cluster registration as in MAMaLGaM-X) when the use of the univariate factorization fails (e.g. on problem BD_2^s).

Figure 6 shows convergence graphs for use of the full-covariance Gaussian. MAMaLGaM-X either performs similar to SDR-AVS-MIDEA or outperforms it. This shows that the BKLM clustering and explicit cluster registration techniques are beneficial and promising in general for multi-objective optimization with mixture-based EDAs.

If the Bayesian factorization or univariate factorization are used, convergence happens faster because a much smaller population size can be used. For our test suite, a similar success rate is even obtained, with the exception of BD_2^s. On this problem, slower convergence is obtained using Bayesian factorizations and the optimum cannot be found using univariate factorizations. Although it concerns only a single test problem here, this does illustrate the important fact to keep in mind that not all problems can be solved efficiently without taking dependencies into account, which is also in accordance with findings for discrete MO problems [12]. Examining this importance in the light of more practical or even real-world problems is however an important topic of future research. Also, although the problems used here have dependencies between the problem variables, i.e. because of the Rosenbrock problem in BD_1 and BD_2^s, these dependencies are of low order. Using Bayesian factorizations rather than a full covariance matrix the Rosenbrock function can be optimized more efficiently. Moreover, using AMS, the optimum can be found even with the univariate factorization, albeit it less efficiently. For this reason the optimum of BD_1 and BD_2^s can be found using MAMaLGaM-X$^+$ for all variants of dependency modelling. On the one hand this demonstrates the potential of the proposed SO-MO combination. On the other hand, this stresses even more the importance of testing the influence of high-order dependency modelling on more practical or even real-world MO problems.

Overall, MAMaLGaM-X$^+$ performs the best. While requiring only marginally more effort in terms of function evaluations, good approximations can be found in all runs on all problems. The exception is problem EC_4, where all tested MOEDAs almost always fail. This problem is highly multi-modal. Furthermore, the optima of the EC problems lie on the boundary of the search space. Finally, we note

Full covariance matrix									
	BD_1	BD_2^s	GM_1	GM_2	EC_1	EC_2	EC_3	EC_4	EC_6
Without AMS									
SDR-AVS-MIDEA-05	83	0	100	100	0	0	13	0	40
SDR-AVS-MIDEA-10	100	0	100	100	0	0	0	0	6
MAMaLGaM-X-10	93	0	100	100	6	0	0	0	100
MAMaLGaM-X-20	83	3	100	100	0	0	0	3	0
With AMS									
SDR-AVS-MIDEA-05	96	3	100	100	100	83	86	0	100
SDR-AVS-MIDEA-10	96	3	100	100	0	0	0	0	100
MAMaLGaM-X-10	100	3	100	100	100	10	93	0	100
MAMaLGaM-X-20	100	63	100	100	100	100	93	3	100
MAMaLGaM-X$^+$-10	100	100	100	100	100	100	96	0	100
MAMaLGaM-X$^+$-20	100	100	100	100	100	100	100	0	100
Bayesian factorization									
	BD_1	BD_2^s	GM_1	GM_2	EC_1	EC_2	EC_3	EC_4	EC_6
Without AMS									
SDR-AVS-MIDEA-05	90	3	100	100	100	100	100	0	100
SDR-AVS-MIDEA-10	100	86	100	100	100	100	100	0	100
MAMaLGaM-X-10	43	0	100	100	100	100	100	0	100
MAMaLGaM-X-20	80	0	100	100	100	100	90	3	100
With AMS									
SDR-AVS-MIDEA-05	86	10	100	100	100	100	100	0	100
SDR-AVS-MIDEA-10	100	100	100	100	100	100	100	0	100
MAMaLGaM-X-10	100	40	100	100	100	100	100	3	100
MAMaLGaM-X-20	100	96	100	100	100	100	100	6	100
MAMaLGaM-X$^+$-10	100	100	100	100	100	100	100	0	100
MAMaLGaM-X$^+$-20	100	100	100	100	100	100	100	0	100
Univariate factorization									
	BD_1	BD_2^s	GM_1	GM_2	EC_1	EC_2	EC_3	EC_4	EC_6
Without AMS									
SDR-AVS-MIDEA-05	0	0	100	100	100	100	80	0	100
SDR-AVS-MIDEA-10	0	0	100	100	100	100	100	0	100
MAMaLGaM-X-10	0	0	100	100	100	55	93	0	100
MAMaLGaM-X-20	0	0	100	100	100	100	46	0	100
With AMS									
SDR-AVS-MIDEA-05	6	0	100	100	100	100	70	0	100
SDR-AVS-MIDEA-10	0	0	100	100	100	100	100	0	100
MAMaLGaM-X-10	86	0	100	100	100	39	96	0	100
MAMaLGaM-X-20	100	0	100	100	100	100	100	0	100
MAMaLGaM-X$^+$-10	30	96	100	100	100	100	100	0	100
MAMaLGaM-X$^+$-20	100	96	100	100	100	100	100	0	100

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

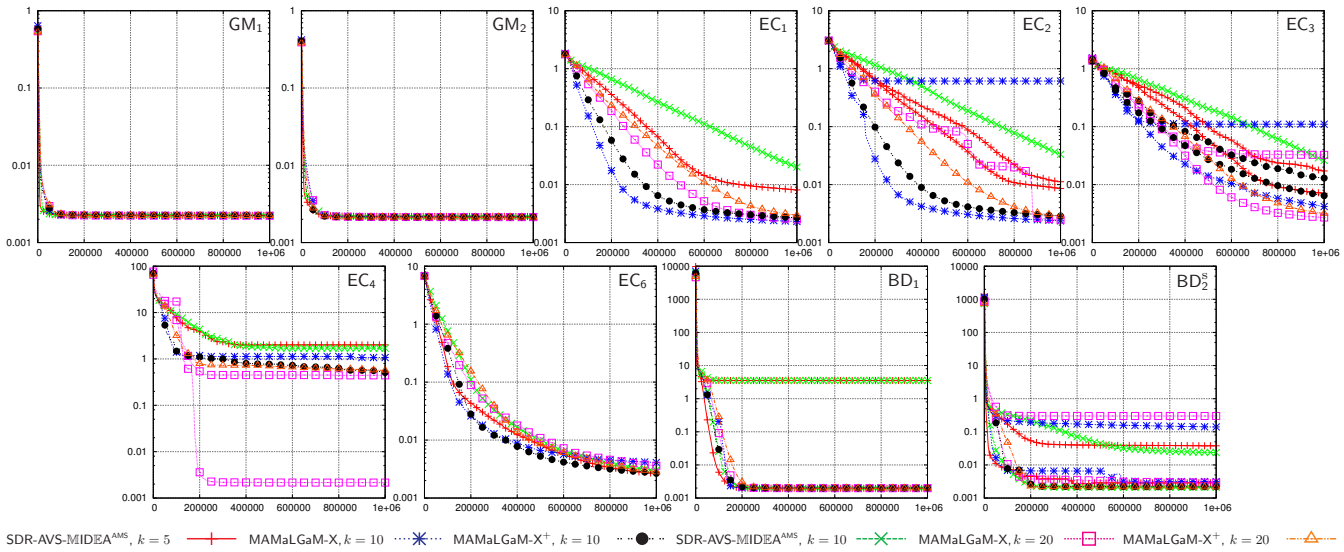

Figure 6: Average performance of various MOEDAs on all problems, estimating full covariance matrices in each cluster. Horizontal axis: number of evaluations (both objectives per evaluation). Vertical axis: $D_{\mathcal{P}_F \to \mathcal{S}}$. For each algorithm averages are shown both for succesful runs and unsuccesful runs, giving double occurrences of lines if some runs were unsuccesful.

that the $< 100\%$ successrate of the univariately-factorized MAMaLGaM-X$^+$ is only due to the limit of 10^6 evaluations, around which budget the MOEDA is always near the required $D_{\mathcal{P}_F \to \mathcal{S}}$ score of 0.01.

9. SUMMARY AND CONCLUSIONS

To find good approximations of the optimal Pareto front, continued pressure toward finding improvements is required. If the Pareto front spreads fast, this pressure can be hard to maintain, especially in the real-valued case where infinitely many solutions are available. As many solutions of a similar quality are then selected, a MOEDA can easily converge prematurely due to overfitting solutions of that quality, i.e. a contour in the fitness landscape. Enlarging the capacity of the probabilistic model via mixture distributions and the modelling of dependencies only increases the probability that such overfitting can occur, contrary to what is commonly expected from EDAs when employing more complex distributions. The techniques described in this paper reduce this risk substantially and effectively. Moreover, using the proposed BKLM clustering technique any EDA can be extended to a mixture-based version straightforwardly. In future work we shall use this approach to further study the convergence of MOEDAs in discrete search spaces. We shall also investigate the use of incremental learning methods to reduce the required number of solutions per cluster. Especially in combination with many clusters, this can potentially lead to large performance improvements.

10. REFERENCES

[1] P. A. N. Bosman. On empirical memory design, faster selection of Bayesian factorizations and parameter-free Gaussian EDAs. In G. Raidl et al., editors, *Proc. of the Genetic and Evolutionary Comp. Conf. - GECCO-2009*, pages 389–396, New York, New York, 2009. ACM Press.

[2] P. A. N. Bosman and D. Thierens. Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms. *International J. of Approx. Reasoning*, 31(3):259–289, 2002.

[3] P. A. N. Bosman and D. Thierens. Adaptive variance scaling in continuous multi-objective estimation-of-distribution algorithms. In D. Thierens et al., editors, *Proc. of the Genetic and Evol. Comp. Conf. - GECCO-2007*, pages 500–507, New York, New York, 2007. ACM Press.

[4] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer-Verlag, Berlin, 2007.

[5] M. Gallagher and M. Frean. Population-based continuous optimization, probabilistic modelling and mean shift. *Evolutionary Computation*, 13(1):29–42, 2005.

[6] C. González, J. A. Lozano, and P. Larrañaga. Mathematical modelling of UMDAc algorithm with tournament selection. behaviour on linear and quadratic functions. *International Journal of Approximate Reasoning*, 31(3):313–340, 2002.

[7] J. Grahl, S. Minner, and F. Rothlauf. Behaviour of UMDAc with truncation selection on monotonous functions. In D. Corne et al., editors, *Proceedings of the IEEE Congress on Evolutionary Computation - CEC-2005*, pages 2553–2559, Piscataway, New Jersey, 2005. IEEE Press.

[8] N. Hansen. The CMA evolution strategy: a comparing review. In J. A. Lozano et al., editors, *Towards a New Evolutionary Computation. Advances in Estimation of Distribution Algorithms*. Springer–Verlag, Berlin, 2006.

[9] K. Harada, J. Sakuma, and S. Kobayashi. Local search for multiobjective function optimization: Pareto descend method. In M. Keijzer et al., editors, *Proc. of the Genetic and Evolutionary Computation Conf. - GECCO-2006*, pages 659–666, New York, New York, 2006. ACM Press.

[10] T. Hiroyasu, M. Nishioka, M. Miki, and H. Yokouchi. Discussion of search strategy for multi-objective genetic algorithm with consideration of accuracy and broadness of Pareto optimal solutions. In X. Li et al., editors, *Simulated Evolution and Learning - SEAL-2008*, pages 339–348, Berlin, 2008. Springer–Verlag.

[11] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

[12] M. Pelikan, K. Sastry, and D. E. Goldberg. Multiobjective hBOA, clustering and scalability. In H.-G. Beyer et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2005*, pages 663–670, New York, New York, 2005. ACM Press.

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