A Novel Population-based Multi-Objective CMA-ES and the Impact of Different Constraint Handling Techniques

Sílvio Rodrigues
Delft University
of Technology (TUDelft)
Delft, The Netherlands
s.m.fragosorodrigues
@tudelft.nl

Pavol Bauer
Delft University
of Technology (TUDelft)
Delft, The Netherlands
p.bauer@tudelft.nl

Peter A.N. Bosman
Centrum Wiskunde
& Informatica (CWI)
Amsterdam, The Netherlands
peter.bosman@cwi.nl

ABSTRACT

The Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) is a well-known, state-of-the-art optimization algorithm for single-objective real-valued problems, especially in black-box settings. Although several extensions of CMA-ES to multi-objective (MO) optimization exist, no extension incorporates a key component of the most robust and general CMA-ES variant: the association of a population with each Gaussian distribution that drives optimization. To achieve this, we use a recently introduced framework for extending population-based algorithms from single- to multi-objective optimization. We compare, using six well-known benchmark problems, the performance of the newly constructed MO-CMA-ES with existing variants and with the estimation of distribution algorithm (EDA) known as iMAMaLGaM, that is also an instance of the framework, extending the single-objective EDA iAMaLGaM to MO. Results underline the advantages of being able to use populations. Because many real-world problems have constraints, we also study the use of four constraint-handling techniques. We find that CMA-ES is typically less robust to these techniques than iAMaLGaM. Moreover, whereas we could verify that a penalty method that was previously used in literature leads to fast convergence, we also find that it has a high risk of finding only nearly, but not entirely, feasible solutions. We therefore propose that other constraint-handling techniques should be preferred in general.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Global optimization

General Terms

Algorithms, Performance, Experimentation

Keywords

CMA-ES, Estimation-of-Distribution Algorithms, Mixture Distributions, Multi-Objective Optimization

1. INTRODUCTION

Many real-world problems have more than one conflicting objective that need to be optimized simultaneously. Moreover, many problems require taking a black-box optimization (BBO) perspective, i.e. assume that (virtually) nothing is known about the problem at hand (e.g. complex simulation based real-world models). Studying and understanding algorithms to tackle optimization problems under such conditions is therefore important.

For single-objective (SO) real-valued optimization problems, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is a well-known, state-of-the-art algorithm, especially in black-box settings. Several multi-objective (MO) extensions of CMA-ES exist [1, 2, 3, 4]. However, none incorporates a key component of the most robust and general CMA-ES variant [5]: the association of a population with each Gaussian distribution to drive optimization. In the first MO extension of CMA-ES [1], a $(1 + \lambda)$-MO-CMA-ES is introduced. In this approach, each parent generates one or more offspring solutions. Thereafter, parents and offspring solutions are selected based on non-dominated sorting. The search strategy parameters are updated based on a direct comparison between offspring and respective parents. However, there is no information-sharing mechanism for different individuals, even if they are in similar search space areas. In the second proposed MO-CMA-ES variant, in each generation only one parent is selected to generate one offspring [2]. The parent may be randomly selected from the entire population or, in a greedy approach, from the subset with the lowest non-dominated rank. In another MO-CMA-ES variant, a notion of continuity of the optimization problem was assumed [3]. Successful offspring then update their covariance matrix taking into consideration information not only from their parent but also from all the successful offspring solutions of that generation. In the latest MO-CMA-ES variant, a new success criterion was introduced: an offspring is considered successful if it is present in the next parent population. This criterion is broader since offspring are not compared only with their respective parent, but rather, how well they perform in the entire population is also evaluated.

All existing MO-CMA-ES extensions are based on the $(1 + \lambda)$-MO-CMA-ES which use populations of size one. Using larger populations could be beneficial however, since an increased spread and coverage of the Pareto front would be expected. Moreover, if all the population information content is used to drive optimization, the algorithm is less prone to getting stuck in sub-optimal local minima [1].

Many real-world problems have constraints, making the performance of BBO algorithms under different constraint-handling techniques important. All MO-CMA-ES variants previously introduced use a penalty term to handle box constraints. This approach lead to fast convergence speeds for certain benchmark problems [1, 2, 3, 4]. However, this approach has drawbacks. For example, it only performs well with box constraints since these, differently from general problem constraints, allow an easy mapping to the feasible space. Furthermore, infeasible solutions may be accepted in the elitist archive with this constraint handling technique.

The main objectives of this paper stem from the fact that all existing MO-CMA-ES variants use populations of size one and that only the penalty approach was used to handle constraints. Our first goal is to study the benefits of having a population-based MO-CMA-ES. To do so, we study a combination between a multi-objective optimization framework that was recently introduced [6] with the most general SO version of CMA-ES [5]. Our second goal is to assess the performance and robustness of the previously introduced MO-CMA-ES variants, the novel population-based MO-CMA-ES and the iMAMaLGaM algorithm [6] under different and more general constraint handling techniques.

The remainder of this paper is organized as follows: Section 2 briefly introduces the key concepts in MO optimization. Then, Section 3 describes the most popular MO-CMA-ES variants. Subsequently, in Section 4 a recently introduced general framework for extending population-based algorithms from single- to multi-objective optimization is described. Thereafter, iMAMaLGaM is described in Section 5. The novel population-based MO-CMA-ES is introduced in Section 6. Subsequently, the different constraint-handling techniques are briefly discussed in Section 7. The performance of the algorithms is tested in Section 9 on 6 benchmark problems and with different constraint-handling techniques. A discussion of the results is given in Section 10, while final conclusions are presented in Section 11.

2. MULTI-OBJECTIVE OPTIMIZATION

We assume to have m objective functions $f_i(\boldsymbol{x})$, $i \in \{1, 2, \ldots, m\}$ and, without loss of generality, we assume that the goal is to *minimize* all objectives. A solution $\boldsymbol{x}^1$ is said to (Pareto) *dominate* a solution $\boldsymbol{x}^2$ (denoted $\boldsymbol{x}^1 \succ \boldsymbol{x}^2$) if and only if $f_i(\boldsymbol{x}^1) \le f_i(\boldsymbol{x}^2)$ holds for all $i \in \{1, 2, \ldots, m\}$ and $f_i(\boldsymbol{x}^1) < f_i(\boldsymbol{x}^2)$ holds for at least one $i \in \{1, 2, \ldots, m\}$. A *Pareto set* of size n is a set of solutions $\boldsymbol{x}^j$, $j \in \{1, 2, \ldots, n\}$ for which no solution dominates any other solution, i.e. there are no $j, k \in \{1, 2, \ldots, n\}$ 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 $\boldsymbol{f}(\boldsymbol{x}^j)$, $j \in \{1, 2, \ldots, n\}$. A solution $\boldsymbol{x}^1$ is said to be *Pareto optimal* if and only if there is no other $\boldsymbol{x}^2$ such that $\boldsymbol{x}^2 \succ \boldsymbol{x}^1$ 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 $\mathcal{P_S}$ and the optimal Pareto front by $\mathcal{P_F}$.

3. MO-CMA-ES

The four MO-CMA-ES variants used for testing are briefly explained here. For more in-depth explanations, the interested reader is referred to the original works [1, 2, 3, 4].

The so-called generational MO-CMA-ES was the first attempt to construct a MO CMA-ES [1]. A $(1 + 1)$-CMA-ES was used to this end. The idea is to maintain a population of λ_{MO} elitist $(1 + 1)$-MO-CMA-ES instances which are subjected to selection based on non-dominated sorting. The hypervolume measure, defined as the Lebesgue measure of the union of hypercubes in the objective space [7], was used as second level sorting criterion to rank individuals with the same level of non-dominance.

The so-called Steady-State Selection MO-CMA-ES was the second attempt. Although, this approach is very similar to the generational variant, in each generation, only one parent solution generates one offspring [2]. This variant is computationally more demanding than the generational variant since all the necessary computations to update the parent pool are performed for each new offspring.

The so-called Recombination MO-CMA-ES [3] was the first attempt to transfer information between successful offspring. Each individual adapts its own covariance matrix considering not only its parent but also all successful offspring generated in the same generation. The underlying idea is that individuals that are close in the search space are likely to be similar if a notion of continuity is assumed. Nonetheless, the influence of successful offspring is weighted by their distance to the solution to be updated.

In the Improved Step MO-CMA-ES [4], an offspring is said to be successful if it is present in the next parent population. This not only reduces computational complexity, but it also improves performance since the success criterion is broader since a solution is compared to a set of solutions instead of solely with the individual from which it originated.

4. MO OPTIMIZATION FRAMEWORK

A general framework for extending population-based algorithms from single- to multi-objective optimization was introduced in [6]. An overview of the multi-objective optimization framework is depicted in Figure 1. Next, a description of each step of the framework is given.
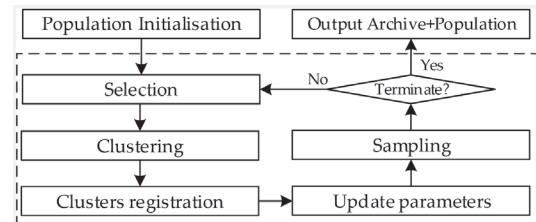

Figure 1: Flowchart of the multi-objective optimization framework.

4.1 Selection

Given a population of size n, $\lfloor \tau n \rfloor$ ($\tau \in [\frac{1}{n}, 1]$) solutions with the lowest domination ranks are selected. From the rank that crosses the boundary of $\lfloor \tau n \rfloor$ solutions a subset must be chosen. For this, the same nearest-neighbor heuristic is used as for selecting cluster leaders (see below).

4.2 Clustering

A nearest-neighbor heuristic is used to select $k - m$ cluster leaders, that are spread as well as possible, from the selection set. 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 first leader and the most distant solution 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-neighbor distance. This procedure is repeated until all the necessary cluster leaders are selected.

Next, the distance from each solution in the selection set to the cluster leaders is computed. These leaders serve as initial cluster centroids in the subsequent application of the k-means clustering algorithm. After running k-means clustering, for each cluster, the c closest solutions are to the final cluster centroid assigned to it, ensuring that each cluster consists of exactly c solutions.

Because the final assignment is done independently for each cluster, some solutions may be assigned to multiple clusters whereas other solutions may not be assigned at all. The probability of this happening can be reduced by increasing the probability that the clusters will overlap by setting $c > \frac{1}{k}\lfloor \tau n \rfloor$. Specifically, we use $c = \frac{2}{k}\lfloor \tau n \rfloor$ as proposed in [6], 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 solutions faster.

To ensure that spatial separation of the search bias is obtained in the objective space, clustering is performed on the basis of objective values. Specifically, the distances are 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 are computed from all selected solutions. A point in objective space $f(\boldsymbol{x})$ is then scaled linearly to the observed ranges, i.e. $(f(\boldsymbol{x}) - f_i^{\min})/(f_i^{\max} - f_i^{\min})$.

4.3 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) [6] approach in estimation of distribution algorithms (EDAs) and the evolution path and the estimation of the covariance matrix based on the mean shift in CMA-ES [5]. The performance of these mechanisms strongly depends on a correlation between the solution sets in subsequent generations. By re-applying clustering each generation however, in principle, there is no spatial relation between clusters in subsequent generations, i.e. the final enumeration of clusters does not guarantee that cluster i in generation $t - 1$ is near cluster i in generation t.

To obtain a smooth evolution of clusters and their paths over subsequent generations, a registration between clusters in subsequent generations that minimizes the distance of the matched clusters is computed [6]. To do so, a matching of clusters is computed in subsequent generations that minimizes the distance between the matched clusters. Firstly, the distances between clusters in generations t and $t-1$ are calculated. The distance between two clusters is taken to be the sum of the smallest pair-wise distance between the cluster's solutions. Furthermore, the cluster distances are computed between the clusters in generation t and between the clusters in generation $t - 1$. Then, $r \leq k$ clusters, from each generation, are repeatedly selected to be registered. To this end, first the two still-unregistered clusters in generation t are determined that are the farthest apart. One of

these two clusters is randomly selected, as well, as the still-unregistered cluster in generation $t - 1$ that is closest to it. The $r - 1$ nearest neighbors 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. The permutation for which the sum of the distances between the matched clusters is minimal, is selected. Subset registration is 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, exhaustive optimization by enumeration can only be done for small values of r.

4.4 Parameter estimation

After explicit cluster registration, depending on the optimization algorithm being combined with the framework, the parameters that span multiple generations are updated.

4.5 Elitism

An elitist archive is maintained with all currently non-dominated solutions. If the objectives are real-valued, there will be infinitely many non-dominated solutions possible. To prevent the archive from growing to extreme sizes, 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.

Keeping elitist solutions in the population can improve convergence. Therefore, each solution in the elitist archive is associated with its nearest cluster. The distance is computed between the cluster mean to the archive solution. 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-neighbor heuristic is used as in selection. Finally, each cluster generates equally many solutions. Depending on how many elitist solutions were copied to the population, at least $n - \lfloor \tau n \rfloor$ new solutions are generated.

4.6 New aspects

In this work several changes are made to the optimization framework presented in [6]. The modifications are as follows:

- The k-means algorithm is not used because the clusters means tend to drift "inwards" in the selection set, leading to reduced search effort in the vicinity of the Pareto extremes. Instead, the clusters are grown directly around the leaders chosen from the selection set;

- The previous framework included m external, independent SO optimizers whose best solutions were infused into the MO population. Instead, m clusters are added to the MO population that are constructed by selecting solutions from the population in a single-objective manner. This effectively keeps SO pressure on the Pareto extremes while ensuring that this pressure in terms of solutions overlaps with the MO population.

With external SO optimizers there is a larger probability of the Pareto extremes becoming fully disconnected from the population and Pareto front as governed by the MO population;

- An adaptive gridding technique was employed for the elitist archive [8]. The main advantage of this is that only a target size for the elitist archive needs to be specified while all the gridding details are automatically handled. Initially, all solutions are accepted in the archive. After an upper bound size is crossed, a grid adaptation process is triggered to ensure that the number of elitist solutions stays in the vicinity of the target size. This method provides the same convergence guarantee as the static grid approach, only potentially losing convergence properties when the grid is redefined. Furthermore, as shown in [8] grid adaptations do not occur often during a run and no loss in performance was observed in experiments;

- Solutions that share an elitist hypercube and do not dominate the already existing solution found there but further extend the front are accepted. Such solutions were previously discarded. This feature guarantees that the front is always extended since the granularity of the grid plays no part in the process.

5. IMAMALGAM

The following is a brief description of the MO-EDA that we compare our results to. The MO-EDA follows the above framework and was previously called iMAMaLGaM-X^{+}, indicative of being an extension of an earlier version called iMAMaLGaM-X and the SO-EDA iAMaLGaM (the internal mechanisms of which are used in each cluster in the multi-objective framework). For a more in-depth explanation the interested reader is referred to literature [6, 9]. In the remainder however, for simplicity, we shall generally use the acronym iMAMaLGaM to indicate the framework-leveraged MO version of iAMaLGaM. Moreover, we shall use iMAMaLGaM$^{\text{Old}}$ to denote the use of the previously existing framework without the above mentioned new aspects.

After clustering and 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 [6]. The AMS is computed as the difference between the means of subsequent generations, i.e. $\hat{\boldsymbol{\mu}}^{\text{Shift}}(t) = \hat{\boldsymbol{\mu}}(t) - \hat{\boldsymbol{\mu}}(t-1)$. If the AMS-altered solutions violate box constraints, a shrinking factor is used to keep the solution feasible. Specifically $\alpha 100\%$ of the newly sampled solutions are moved in the direction of the AMS: $\boldsymbol{x} \leftarrow \boldsymbol{x} + 2\hat{\boldsymbol{\mu}}^{\text{Shift}}(t)$. The rationale is that solutions so changed 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 $\alpha = \frac{1}{2}\tau \frac{n}{n - n^{\text{elitist}}}$.

By estimating the 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 no help

much as it 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 in 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.

In iMAMaLGaM memory decay is used in estimating the covariance matrix and the AMS in generation t as follows:

$$\hat{\boldsymbol{\Sigma}}(t) = (1-\eta^{\boldsymbol{\Sigma}})\hat{\boldsymbol{\Sigma}}(t-1) + \eta^{\boldsymbol{\Sigma}}\frac{1}{|\boldsymbol{\mathcal{S}}|}\sum_{i=0}^{|\boldsymbol{\mathcal{S}}|-1}(\boldsymbol{\mathcal{S}}_i - \hat{\boldsymbol{\mu}}(t))(\boldsymbol{\mathcal{S}}_i - \hat{\boldsymbol{\mu}}(t))^T$$

$$\hat{\boldsymbol{\mu}}^{\text{Shift}}(t) = (1-\eta^{\text{Shift}})\hat{\boldsymbol{\mu}}^{\text{Shift}}(t-1) + \eta^{\text{Shift}}(\hat{\boldsymbol{\mu}}(t) - \hat{\boldsymbol{\mu}}(t-1)).$$

Values for the learning-rate parameters $\eta^{\boldsymbol{\Sigma}}$ and η^{Shift} were determined empirically [9]. The cluster sizes computed from the selected solutions were set according to guidelines from recent literature on SO [9]: $n^{cluster} = \tau 10 l^{0.5}$.

6. POPULATION-BASED MO-CMA-ES

A novel population-based MO-CMA-ES was obtained by combining the most robust and general CMA-ES variant [5], $(\mu/\mu_W, \lambda)$-CMA-ES, with the MO optimization framework. For the solutions in the single-objective clusters, domination ranks are assigned in a linear way, with the best solution getting assigned rank one, similar to the $(\mu/\mu_W, \lambda)$-CMA-ES strategy. For the remaining clusters, the ranks are determined through a non-dominated sorting scheme per cluster.

The size of each individual cluster is an important parameter. Since recombination may smooth the effect of selecting erroneous solutions, increasing solely λ is inferior to randomly generating new solutions, whereas increasing both μ and λ are superior [5]. The cluster size was defined as two times larger than the default value of the population size of the single objective $(\mu/\mu_W, \lambda)$-CMA-ES:

$$\text{cluster size} = \lfloor 2\tau \times (4 + \lfloor 3 \ln n \rfloor) \rfloor \qquad (1)$$

This was done to maintain the cluster size similar to the population size given as guideline [5] since $\tau = 0.5$. Moreover, all the remaining internal parameters of CMA-ES were also kept the same as the standard guidelines [5].

7. CONSTRAINT HANDLING

Four different constraint handling techniques were implemented in order to assess their impact on the performance of the algorithms under study:

- Resampling [10]: a newly generated solution is resampled if it lies outside the variables boundaries. After 100 trials, a new solution is randomly generated;

- Constraint domination: for each constraint, a measure of the amount by which the constraint is violated, is used. The total constraint-violation value is the sum over all constraints. For box constraints, the constraint-violation value is the distance to the closest feasible solution. Solutions are then ranked using constraint-domination ranking whereby a solution with a smaller constraint violation is always preferred. If two solutions are both feasible, the one with the lowest domination rank is performed;

- Penalty term: if a solution violates box constraints, the closest feasible solution is evaluated. A penalty

term is then added to the fitness value. This has been the constraint handling technique of choice in previous MO-CMA-ES tests [1, 2, 3, 4].

- Global Competitive Ranking [11]: this ranking approach aims to create a balance between objective values and constraint violations, allowing both feasible and infeasible solutions to preside in the population. In this way, the algorithm may approach the optimal front from both the feasible and the infeasible sides, leading, in theory, to faster convergence speeds. A weighing factor, P_f, sets the relevance of one ranking scheme compared to the other. If $P_f = 0$, this scheme behaves as constraint domination, whereas if $P_f = 1$ the constraint violations are neglected and ranking is purely based on fitness. A value of $P_f = 0.45$ was used since it is known to provide good results [11].

The boundary repair technique, by which variables are set to their closest bound, in case they fall outside of the allowed range, was not considered. Key reason is that some of the artificial benchmark functions that we consider have optimal Pareto fronts which are obtained by setting all variables to one of the bounds while only one variable defines the front size and shape. In this way, if boundary repair is carried, infeasible solutions may be altered and placed automatically in the optimal front, biasing the results.

8. EXPERIMENTS

For all benchmark problems the algorithms were allowed one million function evaluations (per objective). However, if the distance to the front, as measured using the performance indicator introduced in 8.2, does not improve at least 10^{-10} after 50 generations, the algorithm is stopped and convergence is deemed to be too slow.

Twenty clusters were used in iMAMaLGaM since it was previously found to provide good results [6]. For registration, two subsets of seven clusters ($r = 7$) and one of six clusters ($r = 6$) were used. A similar number of clusters was used for the population-based MO-CMA-ES. The cluster sizes of the population-based MO-CMA-ES were 10 and 14 for problem dimensions of 10 and 30, respectively. For iMAMaLGaM the cluster sizes were 11 and 19, respectively.

For all the MO-CMA-ES instances from literature, the population size was set to $\lambda = 20$ and the offspring number was set to one per population. The population size was chosen to lead to a similar number of estimated distributions, per generation, in all algorithms. In the case of the steady-state algorithm, the variant where all the population members are potential parents was selected [2].

The selection percentile was set to $\tau = 0.35$ for iMAMaLGaM. For the population-based MO-CMA-ES a value of $\tau = 0.5$ was used since it is the SO-$(\mu/\mu_W, \lambda)$-CMA-ES guideline, which corresponds to a maximal step-size for a given λ [12].

The desired archive size was set to 1000 solutions.

8.1 Test suite

The definitions of the problems in our test suite are presented in Table 1. We used the well-known problems ZDT_i, $i \in \{1, 2, 3, 6\}$. The IRs of the ZDT_i problems are also constraints. For more details about these functions, see [13]. The final two problems are labeled BD_i, $i \in \{1, 2\}$ and

were taken from [6]. Both problems make use of the Rosenbrock 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 since the objective functions overlap in all variables instead of only in x_0. Furthermore, 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 $D_{\mathcal{P}_F \to \mathcal{S}}$ indicator (explained in the next Section) corresponds to a similar front quality on all problems.

Name	Objectives	IR
ZDT_1	$f_0 = x_0, \quad f_1 = \gamma\left(1 - \sqrt{f_0/\gamma}\right)$ $\gamma = 1 + 9\left(\sum_{i=1}^{l-1} x_i/(l-1)\right)$	$[0;1]^{30}$ $(l = 30)$
ZDT_2	$f_0 = x_0, \quad f_1 = \gamma\left(1 - (f_0/\gamma)^2\right)$ $\gamma = 1 + 9\left(\sum_{i=1}^{l-1} x_i/(l-1)\right)$	$[0;1]^{30}$ $(l = 30)$
ZDT_3	$f_0 = x_0, \quad f_1 = \gamma\left(1 - \sqrt{f_0/\gamma} - (f_0/\gamma)\sin(10\pi f_0)\right)$ $\gamma = 1 + 9\left(\sum_{i=1}^{l-1} x_i/(l-1)\right)$	$[0;1]^{30}$ $(l = 30)$
ZDT_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.

8.2 Measuring performance

We consider the Pareto set that can be computed from the elitist archive combined with the population upon termination to be the approximation set, denoted $\mathcal{S}$. To measure performance, the $D_{\mathcal{P}_F \to \mathcal{S}}$ performance indicator is computed. This performance indicator computes the average distance over all points in the optimal Pareto front $\mathcal{P}_F$ to the nearest point in $\mathcal{S}$:

$$D_{\mathcal{P}_F \to \mathcal{S}}(\mathcal{S}) = \frac{1}{|\mathcal{P}_F|} \sum_{\boldsymbol{f}^1 \in \mathcal{P}_F} \min_{\boldsymbol{f}^0 \in \mathcal{S}} \{d(\boldsymbol{f}^0, \boldsymbol{f}^1)\} \qquad (2)$$

where $\boldsymbol{f}$ is a point in objective space and $d(\cdot, \cdot)$ computes Euclidean distance. A smaller $D_{\mathcal{P}_F \to \mathcal{S}}$ value is preferable and a value 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 $\mathcal{S}$ 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 $\mathcal{P}_F$ 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 $D_{\mathcal{P}_F \to \mathcal{S}}$ indicator corresponds to fronts that are close to the optimal Pareto front.

9. RESULTS

The results are averaged over 100 independent runs and shown in Figure 2. For each algorithm, averages are shown both for successful and unsuccessful runs, giving double occurrences of lines if some runs were unsuccessful. All combinations of algorithms and constraint-handling techniques are shown, except for iMAMaLGaM$^{\text{Old}}$ for which results are only shown with resampling. The differences between the old and new versions of iMAMaLGaM have a positive, but small effect in terms of convergence, leading to similar differences when other constraint-handling techniques are used and making resampling sufficiently illustrative of the impact of the new aspects described in Section 4.6.

9.1 Unconstrained problems

None of the MO-CMA-ES implementations from literature were able to solve unconstrained problem BD_2^s as the algorithms were able to find only one of the Pareto extremes, solving the Sphere function. Because solving Sphere is much simpler than solving Rosenbrock, the population is pulled quickly toward one end of the Pareto front. Due to the lack of pressure to extend the Pareto front at the edges, i.e. pressure toward improving individual objectives as is present in the MO framework from which population-based MO-CMA-ES and the iMAMaLGaM variants are derived, it is hard for existing MO-CMA-ES implementations to find the other Pareto extreme, resulting in a very low convergence speed which led to an early stop of the optimization run. The population-based MO-CMA-ES and the iMAMaLGaM variants did solve BD_2^s.

9.2 Constrained problems

Besides BD_2^s, all benchmark functions are constrained due to the existence of box constraints.

9.2.1 Resampling

None of the MO-CMA-ES algorithms were able to converge to the optimal Pareto front of any of the benchmark functions with this constraint-handling technique. An important factor here is that MO-CMA-ES does not have the iMAMaLGaM mechanism that employs a shrinking factor when AMS is applied to better approach constraints. Consequently, MO-CMA-ES has intrinsic difficulties to approach variable boundaries without triggering resampling, whereby a randomly generated solution is created and all its internal parameters are set to guideline values. The knowledge previously gathered during the optimization run is discarded, leading to very slow convergence speeds and/or, as in this case, high failure rates.

The population-based MO-CMA-ES demonstrated to be more robust than the MO-CMA-ES instances from literature, obtaining better results on all problems with exception for the BD_1 problem. Moreover, it was able to converge faster than iMAMaLGaM on ZDT_1.

iMAMaLGaM with the new aspects presented convergence speeds at least as good as the old version in all problems. Furthermore, iMAMaLGaM was the only algorithm able to solve all problems.

9.2.2 Constraint domination

No MO-CMA-ES variant obtained successful runs in any of the problems with this constraint-handling technique, as opposed to iMAMaLGaM, which was again able to solve

all problems. iMAMaLGaM did present convergence speeds when compared to the use of resampling. This may explained by the fact that, with constraint domination, the shrinking factor of the AMS technique was not applied since the variables were considered to be unbounded. Moreover, iMAMaLGaM algorithm presented some unsuccessful runs for 3 problems.

9.2.3 Penalty function

All the algorithms converged to the ZDT optimal fronts extremely fast. This is in agreement with previous literature studies [1, 2, 3, 4]. For all ZDT problems, the solutions on the optimal front have their variables set at their lowest boundary, besides the first variable which dictates the size and shape of the optimal front. With the penalty function, the benchmark functions are not evaluated with infeasible solutions but rather the closest feasible solution in parameter space is evaluated. Therefore, this strategy brings advantages to the algorithms to approach the optimal Pareto front in these test problems. However, for all ZDT problems and algorithms, the final elitist archives were filled with mostly infeasible solutions. The reason for this is that all variables have box constraints and, once again, it is only the first variable that sets the size and shape of the Pareto front. In this way, solutions with all negative variables besides the first will be evaluated in the optimal Pareto front, entering, in this way, in the elitist archive.

Regarding then the only other constrained problem in our test suite, BD_1, none of the MO-CMA-ES instances from literature were able to solve it. Differently from the ZDT problems, only the first variable is bounded. In this way, if a solution is infeasible, the evaluated solution will not be placed in the optimal Pareto front since only the first variable is altered. The population-based MO-CMA-ES and iMAMaLGaM variants demonstrated to be more robust and were able to solve all problems. Moreover, the final elitist archives were mostly filled with feasible solutions when BD_1 was solved. The first variable is the only one that has box constraints and it is also the only variable that sets the size and shape of the optimal Pareto front. Infeasible solutions present in the final elitist archive of a successful run, are always mapped only to one of the optimal Pareto extremes. Therefore, the first variable, in the elitist archive solutions, has to vary along its entire range.

9.2.4 Global Ranking Procedure

The results were found to be similar to constraint domination. None of the MO-CMA-ES variants were able to solve any of the benchmark functions. iMAMaLGaM performed better with this constraint-handling technique than with constrained domination on ZDT_1 since all runs were considered successful. On the other hand, similar to constraint domination, some unsuccessful runs for both ZDT_2 and ZDT_3 problems were found.

10. DISCUSSION

The CMA-ES-based MO algorithms that we experimented with, showed a higher susceptibility to non-smooth search-space adaptations created by the constraint domination and global ranking approaches to constraint handling. From the results that we obtained, we can conclude that these constraint-handling techniques do not suit CMA-ES-based approaches very well.
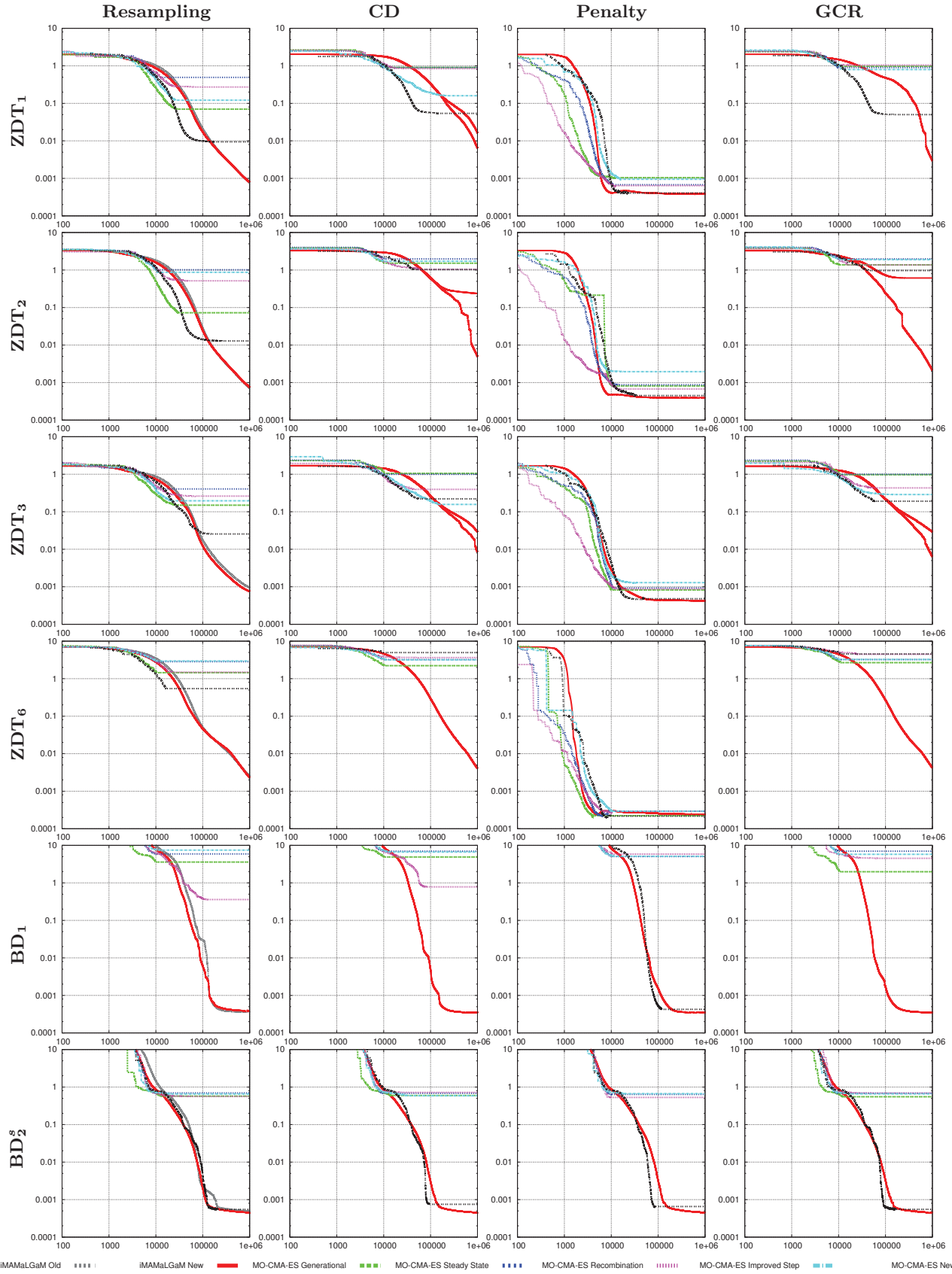
Figure 2: Convergence of all algorithms and all constraint-handling techniques on all problems, averaged over 100 runs. Horizontal axis: number of evaluations (both objectives per evaluation). Vertical axis: $D_{\mathcal{P}_F \to \mathcal{S}}$.

When the penalty term approach was used, all CMA-ES-based algorithms presented very fast convergence speeds for all ZDT problems. This may explained by the fact that this constraint-handling technique does not alter the smoothness of the functions. It is known that CMA-ES is a very powerful SO optimizer for smooth functions [5]. The final elitist archives contained almost only infeasible solutions, however. Although the solutions were slowly moving toward to the feasible area of the search space, it was very difficult to actually enter it. The same holds for iMAMaLGaM, clearly making this a property of the penalty term approach itself. Therefore, if an elitist archive filled with only feasible solutions is desired, this technique is not advised.

The constraint domination and the penalty term techniques enforce different ways in which the search process approaches the optimal Pareto front. With constraint domination, the front is approached from the feasible side, whereas with the penalty term the front is approached from the infeasible side. The global competitive ranking approach is an interesting technique since it allows the algorithms to approach the optimal front from both sides. However, our results do not support any substantial performance improvement over the use of constraint domination alone.

Although in the cluster registration procedure the distance between clusters in subsequent generations is minimized, it is not guaranteed that the cluster means, in subsequent generations, are close to each other. Such phenomenon demonstrated to be counterproductive for the proposed population-based MO-CMA-ES for the benchmark function BD_1. Since the update of the internal parameters, covariance matrix and sigma, depend on the mean shift in subsequent generations, high values may be obtained for these parameters, resulting in erratic behavior. Overall however, the novel population-based MO-CMA-ES was found to be more robust than existing MO-CMA-ES variants.

The population-based MO-CMA-ES presented faster convergence speeds than iMAMaLGaM on smooth functions. This was expected since for single-objective optimization of smooth functions, CMA-ES outperforms iAMaLGaM. However, iMAMaLGaM was found to be able to reach smaller distances to the optimal Pareto fronts. It is therefore debatable which algorithm performs best. If the number of evaluations are the most important criterion and a non-disruptive constraint-handling technique can be designed and used, one might say that population-based MO-CMA-ES is preferable, whereas otherwise iMAMaLGaM can be preferred.

For future work it would be interesting to extend the shrinking factor in the AMS technique to also work with the more general case of constraint domination. One goal in this paper was to test the multi-objective CMA-ES-based algorithms with the same number of distributions as the iMAMaLGaM variants. As future work, it would be interesting to also assess the performance impact if larger population sizes are used. Furthermore, problems with than two objectives should be used in future work and the correlation between the number of objectives and required number of clusters should be assessed.

11. SUMMARY AND CONCLUSIONS

In this paper we introduced a novel population-based MO-CMA-ES algorithm. Experimental results demonstrate that the proposed approach is, in general, more robust when compared to the multi-objective extensions of CMA-ES that were previously introduced in literature.

The performance of several evolutionary multi-objective optimization algorithms under different constraint-handling techniques was assessed. The algorithms based on CMA-ES demonstrated to be very sensitive to the technique applied. On the other hand, iMAMaLGaM showed to be more robust since it was able to solve all problems with all constraint handling techniques.

12. REFERENCES

[1] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, no. 1, pp. 1–28, 2007.

[2] C. Igel, T. Suttorp, and N. Hansen, "Steady-state selection and efficient covariance matrix update in the multi-objective cma-es," in *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization*, Springer, Ed., 2007, pp. 171–185.

[3] T. Voß, N. Hansen, and C. Igel, "Recombination for learning strategy parameters in the mo-cma-es," in *Fifth International Conf. on Evol. Multi-Criterion Optimization (EMO 2009)*. Springer-Verlag, 2009, pp. 155–168.

[4] T. Voß, N. Hansen, and C. Igel, "Improved step size adaptation for the mo-cma-es," in *Proceedings of the 12th annual conference on Genetic and Evol. Comp.* Portland, United States: ACM, 2010, pp. 487–494.

[5] N. Hansen and S. Kern, "Evaluating the cma evolution strategy on multimodal test functions," in *Parallel Problem Solving from Nature (PPSN VIII)*, ser. LNCS, X. Yao *et al.*, Eds., vol. 3242. Springer, 2004, pp. 282–291.

[6] P. A. N. Bosman and T. Alderliesten, "Incremental gaussian model-building in multi-objective edas with an application to deformable image registration," in *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2012.* New York: ACM Press, 2012, pp. 241–248.

[7] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms - a comparative case study," in *Parallel Problem Solving from Nature (PPSN-V).* Springer Berlin Heidelberg, 1998, vol. 1498, pp. 292–301.

[8] N. Luong and P. A. N. Bosman, "Elitist archiving for multi-objective evolutionary algorithms: To adapt or not to adapt," in *Parallel Problem Solving from Nature (PPSN XII).* Berlin: Springer-Verlag, 2012, pp. 72–81.

[9] P. A. N. Bosman, "On empirical memory design, faster selection of bayesian factorizations and parameter-free gaussian edas," in *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2009.* New York: ACM Press, 2009, pp. 389–396.

[10] A. Oyman, K. Deb, and H.-G. Beyer, "An alternative constraint handling method for evolution strategies," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1, 1999.

[11] T. P. Runarsson and X. Yao, "Constrained evolutionary optimization - the penalty function approach," 2002.

[12] H.-G. Beyer, "Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 24, pp. 239–267, 2000.

[13] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, Jun. 2000.