Applied Mathematics and Computation 230 (2014) 65–77

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Repairing normal EDAs with selective repopulation

S. Ivvan Valdez P. *, Arturo Hernández, Salvador Botello

Centre for Research in Mathematics (CIMAT) A.C., C. Jalisco S/N, Mineral de Valenciana, Guanajuato, Gto., C.P. 36000, Mexico

ARTICLE INFO

Keywords:
Normal multivariate EDA
Diversity
Weighted estimators
Evolutionary computation
Global optimization

ABSTRACT

The standard Estimation of Distribution Algorithm (EDA), usually, suffers from premature convergence due to an inherent inability to maintain an adequate variance and to preserve diverse candidate solutions. Normal multivariate EDAs have especially shown a lack of exploration even for convex objective functions. This article introduces several techniques which can be used to enhance the standard Normal multivariate EDA performance. The most important ones are based on (1) pre-selecting the candidate solutions to be evaluated, (2) replacing only a fraction of the population and (3) computing weighted estimators of the mean and covariance matrix. The resulting Normal EDA is competitive with similar approaches, as it is evidenced by statistical comparisons.

1. Introduction

The Estimation of distribution algorithm framework was initially proposed as a probability model of the genetic algorithms (GAs) operators [1,2]. Researchers then discovered several advantages of tackling hard optimization problems in this way, for instance: automated learning of self-adapted parameters, usually computed by using Maximum Likelihood (ML) estimators [1–4]. Additionally, researchers pointed out that EDAs could tackle the learning-linkage problem by using structural learning to estimate variable dependencies [1]. In summary, EDAs intend to capture sufficient information to perform the search, via the estimation of the structure and parameters of a probability distribution. According to this paradigm there are several interesting questions that arise when investigating the EDAs performance:

1. What if the estimation and/or predefined search distribution (the parametric model used to search) has an inherent undesired bias resulting in underexplored regions?
2. What if the search distribution is incapable of representing the structure of the solutions in the selected set?

The first question is not referred to the bias needed to perform the search, but a bias inserted due to the model or the parameter computation, for instance: a research article reports that EDAs reduce the variance naturally in the estimation step by a factor $1-1/N$ in each generation (N = population size) [5], when using ML estimators. Related to the second question, EDAs in continuous search spaces usually assume Normality of the adequate search distribution [1,6]. While the general frameworks of standard EDAs assume that the adequate search distribution, is the underlying distribution of the selected set [7,3]. According to these premises, several conditions must be accomplished in order to guarantee that the optimum will be found: (1) the structure, position, and density of the selected set must contain the adequate and sufficient information to find the optimum, (2) the Normal distribution must be capable of capturing such position, structure and density. If none or only one of these statements is achieved, then, it is quite possible that the algorithm never finds the optimum.

* Corresponding author.
E-mail addresses: ivvan@cimat.mx (S.I. Valdez P.), artha@cimat.mx (A. Hernández), botello@cimat.mx (S. Botello).

0096-3003/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2013.12.081

By structure of the selected set we refer to information in the selected set that can be used to infer the contour-lines and variable relationships in the objective function [8]. Notice that, even if we have captured adequately the structure or variable dependencies (for example by a covariance matrix), additionally we need to pose adequately the probability mass (for example by posing the mean in a Normal distribution). In the case of the Normal distribution, the density is well defined by the covariance matrix and mean, but, for example, in the case of Normal mixtures, each single Normal could be weighted or sampled in agreement with the promising region it covers. That is to say, the density could be defined proportionally to the objective function. In ideal EDAs, the characteristics of: structure, position and density of the selected set are directly related with the fitness/objective function [7]. The selected set is posed in the regions with the best objective value, the structure and density depends on a threshold or a probability of selecting and generating solutions which (ideally) depends on the objective value. In many cases the Normal distribution can not reproduce all the mentioned characteristics. Researchers have deal with the lacks of ML-estimator, basically, by increasing the population diversity[9]. Nevertheless a broad range of algorithms and research articles have dealt with this issue, we group the most representative ones as follows:

Variance scaling [10,6,11,12]. These proposals intend to avoid premature convergence of the Normal EDA. Notice that the Normal EDA only favors the region around the mean, hence, the Normal distribution only can represent single-mode function landscapes with ellipsoidal shapes of the promising region. In addition, the current selected set also has an inherent uncertainty about the function landscape and optimum position. Increasing the variance can help diminish these disadvantages, by covering a wider region, and promote sampling unexplored regions.

Covariance matrix repairing [6]. The covariance matrix repairing schemes can group most of the proposals with a covariance matrix modification. These schemes intend to circumvent numerical problems by detecting and repairing negative or quite small eigen-values in the covariance matrix. They also share similar advantages of variance increasing/scaling.

Using complex models based on Normal mixtures [13–15]. Complex models can better reproduce the structure of the selected set as well as the density associated to each promising region, if there are several of them.

Prediction of moving directions. These approaches belongs to a different kind of enhancements in contrast with the commented above, because the other approaches intend to perform an adequate search by reducing (not so quickly) the exploration area, while the approaches that work with directions often considers that the optimum is distant from the exploration area. For example, Wagner et al. [16], show that, even though, the improvement direction can be inferred from the selected set, the standard Normal EDA (EMNA [1]) samples intensively in a direction with similar objective value than the current population (maximum eigenvalue direction), instead of the direction of maximum improvement (minimum eigenvalue direction). The Eigen-EDA [16] intends to circumvent this problem as well as other approaches, such as the widely studied CMA-ES [17], which summarizes historical improvement directions (evolution path) in the covariance matrix. The anticipated mean-shift [4] computes an improvement direction for the mean, and intends to use this knowledge to accelerate the search process.

For this work we assume that the optimum is inside the exploration region, but not necessarily in the center. This is a reasonable assumption because in many real world problems, we must define the search domain as a priori information. This article presents a proposal of a single-Normal multivariate EDA which tackles all the mentioned issues. For this purpose, we firstly initialize the algorithm with candidate solutions of high diversity, in order to reduce inherent bias and uncertainty of the finite sized selected set. Secondly, we preserve a subset of solutions to capture the function landscape and the promising regions shapes, hence even if the structure of the selected set is not well captured by the Normal distribution, the preserved solutions will be used to reinforce the parameter learning and eventually sample all the promising regions. Thirdly, we intend to compute the adequate density by using weighted estimators, which favors sampling the best known regions. Additionally, our algorithm integrates a mechanism to avoid the evaluation of solutions which are similar to those already known, this is the reason we called our algorithm: Normal based EDA with selective re-population (EDA-SRP).

The article is presented as follows: Section 2 presents the technique to rank solutions according to their diversity, and the methods for the selection and parameter estimation steps. Section 3 presents the Normal EDA proposal. Section 4 presents the numerical experiments and contrasts our proposal with other Normal EDAs. Section 5, presents a general discussion of the algorithm, and contrast the different techniques introduced in this article, in order to analize its individual contribution of each of them. Finally, Section 6 presents the general conclusions and perspectives of future work. Without loss of generality, we refer to a maximization case for all the algorithms and formulas presented in this work.

2. Ranking diverse solutions, selection and weight computation

2.1. The maximin algorithm for ranking diverse solutions

The Maximin selection, similar to the presented in this section, was introduced by Valdez et al. [18] for selecting well spread Pareto fronts in multiobjective evolutionary algorithms. The algorithm is named because an individual is selected according to the maximum of the minimum distances to the already selected set.

The Maximin algorithm is used at the beginning of the search process to select the initial population, and during the search process to rank the solutions. In Algorithm 1, *Rank* represents a diversity value, the most diverse solutions have the minimum values. Our Maximin version uses as input a set of reference points R, which is computed as follows:

1. In the initialization procedure, R contains the points in $\hat{X}$ with the minimum and maximum values in each dimension. It is to say, R contains some points in the decision space, which are taken as the minimum and maximum (in the decision space) found in a large sample $\hat{X}$.
2. During the generations, R is the last selected set, computed according the truncation selection explained in the next section.

The value of -0.1 in line 8, could be any value in $[-1, 0)$. In line 6 we set the distance value to -1 for the individuals already ranked. Thus, the -0.1 is only an indicator of which individuals are not ranked yet.

2.2. Truncation selection

In addition to the Maximin algorithm, we use a truncation selection method which promotes that the selected set has a better objective value each generation, it is described in Algorithm 2.

In Algorithm 2 the first threshold θ^0 is the worst objective value in the initial population, the new threshold for the generation $t+1$ is computed inside the algorithm. The selected set objective values are asked to be greater than θ^t, thus the selected set mean converges to the best individual, due to (most of the time) it is an increasing sequence bounded by the best objective value known so far. In line 1 we obtain the indices of the population decreasingly sorted by its objective value, in line 2 an ϵ value is computed as 10^{-14} times the greatest absolute value among the minimum, maximum, and the difference between minimum and maximum fitness in the population. This ϵ is used to ensure an increasing mean of the selected set instead of a non-decreasing one if *epsilon* $= 0$. In line 3 we truncate half of the population setting k (selected set size) to $n/2$. Lines 4 and 5, remove individuals form the selected set if they are not above the threshold θ^t, ensuring that the selected set size is not less than five percent of the population. Notice that the only case when the mean of the selected set could not be an increasing sequence results when the selected set is reduced to five percent of the population. Finally, line 6 recomputes the threshold θ^t as the worst objective value in the selected set, hence the selected set in the next generation must have greater values than θ^t.

2.3. Computing weights for the parameter estimation

Weighting solutions is an up-to-day topic in EDAs [19]. Evolutionary algorithms, such as recent CMA-ES proposals, have been using weighted estimators, and local meta-models for avoiding the evaluation of candidate solutions [20]. In such proposals, the local meta-model does not depend on the re-weighting process, in contrast, the weights used in our approach for the parameter estimation, are also used for the replacement step, in order to avoid the evaluation of candidate solutions. The weights are computed by simply ranking and normalizing the solutions according to their objective value. The individual with the maximum objective value has the maximum weight $w_{I_1} = 2n/(n(n+1))$, the second maximum value has the second maximum weight $w_{I_2} = 2(n-1)/(n(n+1))$, and so on, until the individual with the minimum objective value with a weight $w_{I_n} = 2/(n(n+1))$. In general, for the decreasingly sorted individuals the weights are given by Eq. 1. Remember that I are the indices of the decreasingly sorted selected set. Using Eq. (1) the sum of all the weights is 1, thus the weights can be seen as a priori probabilities which indicate which data is more or less probable, in this case we are indicating that the best individual is more probable than the worst individual, thus we expect to sample intensively the region around the best individual, because we are indicating explicitly that it is the most probable one.

$$w_{I_i} = \frac{2(n-i+1)}{n(n+1)} \quad \text{for } i = 1..n. \tag{1}$$

3. Normal based EDA with selective re-population (EDA-SRP)

This section introduces a Normal based EDA with selective re-population. It is described in Algorithm 3. In Step 1 a large sample $\hat{X}$, of 6 times n_{rs} times the population size, is uniformly generated. In Step 2 the reference set R is computed as described in Section 2.1. Step 4 performs the Maximim ranking and selects the initial population X of n_{pop} individuals with the maximum Maximin ranking. Step 5 evaluates the initial population, and Step 6 performs the truncation selection according to Algorithm 2. Steps 7 is the weight computation according to Section 2.3. Step 8 and 9 store the selected set and its objective value. Step 10 computes weighted estimators of the mean μ and covariance matrix Σ according to Eqs. 2 and 3.

$$\mu_i = \sum_{j \in I} w_j x_{j,i} \tag{2}$$

$$\Sigma_{i,k} = \sum_{j \in I} (x_{j,i} - \mu_i)(x_{j,k} - \mu_k) w_j \tag{3}$$

Step 12 generates a set of candidate solutions $\bar{X}$ of size $n_{rs} \cdot n_{pop}$. The following steps are especially important for the algorithm in order to maintain and generate diverse solutions: Step 13, computes the Maximin ranking for $\bar{X}$ using the last selected set as reference points. In Steps 14–16, a new ranking $Rank_i^*$ is computed for each $\bar{x}_i \in \bar{X}$, by using the Maximin

ranking of each $\bar{x}_i$ and the value of the weight w_k of the nearest neighbor in the selected set R. This rank measures how good a candidate solution can be, regarding the nearest already-evaluated solution, but at the same time, intends to discriminate solutions similar to those already evaluated by using the Maximin ranking. Steps 17 and 18 select and evaluate the solutions with the greatest $Rank^*$ which have the indices J (of size $n_{pop} - n_S$). Notice that the number of evaluated solutions is less than n_{pop} (between 50 and 95 percent). Steps 19 and 20, insert the last selected set and the new candidate solutions to the population, in order to perform the current selection.

The algorithm requires as problem parameters the number of variables n_{var}, and the vectors of inferior and superior limits respectively x_{inf}, x_{sup}. The possible stopping criteria are: a minimum covariance matrix norm ϵ_{tol}, the maximum number of evaluations max_{eval}, maximum desired value of the objective function f^*_{max}, and as user given parameters: the population size n_{pop} and n_{rs} a resampling rate (see Step 12). Possibly the best stopping criterion is the covariance matrix norm, because it indicates the exploration capacity of the algorithm. The purpose of the other stopping criteria is the comparison with reported results. The next section performs several experiments and comparisons in order to provide evidence of the EDA-SRP performance.

4. Experiments and comparison with other Normal EDAs

In this section we present several comparisons with other Normal based EDAs. The functions used for the comparisons are shown in Table 1.

Comparison with the NichingEDA, classical Normal EDAs and the EigenEDA. The NichingEDA [21] intends to solve the problem of representing the structure or shape of the most promising regions by using Niches, each of them is linked with a probability distribution. Then, some information is shared by using evolutionary operators. Some of the results in the comparison of the NichingEDA are presented here, including unimodal and multimodal objective functions. The results include the $UMDA_c^G$, [1], the $EMNA_{global}$ [1] and the EEDA [16] which is similar to EMNA but with a mechanism to enlarge the variance in the direction of the minimum eigenvalue. Additionally, we compared the EDA-SRP with the $CEGNA_{BGE}$[22], CEGDA[22] and NichingEDA for the Schewfel f_8 problem, this is a challenging problem with multiple local minima. The results for this last comparison are also borrowed from [21].

Comparison settings: The comparison uses the Sphere, Ackley, Rosenbrock and Schwefel problems. In domains of $[-100, 100], [-32, 32], [-30, 30]$ and $[-500, 500]$ respectively. A stopping criterion of the maximum number of evaluations is applied as $5e5, 5e5, 5e6, 4e5$. The population size is 2000 for all the problems.

EDA-SPR settings: The population size is 500 for the Sphere, Ackley and Rosenbrock functions. For the Schwefel f_8 problem the population size is 210. The stopping criterion is the number of function evaluations in order to perform a fair comparison. The resampling rate is 3 for all the experiments, except for the Schwefel which is 4.

Comments about the test: Table 2 shows that the EDA-SRP accurately solves all the problems. In addition, the results are similar or better than the delivered by other approaches.

For the Schwefel problem the results in Table 3 shows that the algorithm has a competitive performance, although the EDA-SRP does not find the optimum, it performs better than the others. Most of the cases the algorithm stops because of

Table 1
Test problems used in comparisons.

Name	Definition	Domain
Rosenbrock	$\sum_{i=1}^{n-1}\left(100(x_i - x_{i+1}^2)^2 + (x_i - 1)^2\right)$	$x \in \{-10, 10\}^n$
Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	$x \in \{-100, 100\}^n$
Ackley	$f(x) = -20 \cdot \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	$x \in \{-32, 32\}^n$
Schwefel f_8	$f(x) = -\sum_i^n x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$x \in \{-500, 500\}^n$

Table 2
Comparison of classical Normal base EDAs (UMDA and EMNA) and enhanced Normal EDAs (EEDA and Niching EDA) with the EDA with selective repopulation (EDA-SRP). Mean and (standard deviation) of the best function value found in 30 runs.

$UMDA_c^G$	$EMNA_{global}$	EEDA	NichingEDA	EDA-SRP
Sphere				
1.653e−43	8.674e−44	1.071e−39	4.507e−9	4.0904e−75
(5.021e−44)	(2.631e−44)	(3.159 e−40)	(1.077e−12)	(2.9307e−75)
Rosenbrock				
8.1958	7.7806	6.3004	4.6258	**0**
(0.0374)	(0.1713)	(0.1259)	(1.6986)	(**0**)
Ackley				
4.4409e−16	4.4409e−16	4.4409e−16	2.4350e−5	1.1102e−15
(0)	(0)	(0)	(1.3137e−5)	(1.4089e−15)

Table 3
Comparison of mixture-Normal base EDAs ($CEGNA_{BGe}$ and CEDNA) and Niching EDA with the EDA with selective repopulation (EDA-SRP). For the Schwefel f_8 problem in 30 dimensions. Best, mean and standard deviation of the best function value found in 30 runs.

Algorithm	Best	Mean	SD
$CEGNA_{BGe}$	−10773.1	−6760.35	2624.33
CEDNA	−8712.31	−5922.54	1892.51
NichingEDA	−8733.51	−8005.39	270.755
EDA-SRP	**−11642.53**	**−10518.53**	1313.21

the number of evaluations, that is to say the norm of covariance matrix is not close to 0, hence, the experiment suggests using a bigger population and number of evaluations in order to improve the quality of the results. As can be observed the parameters are not hard to tune, and/or they are robust in the sense that the same parameter values deliver competitive results for different problems in this test.

The following subsection analyzes the EDA-SRP general aspects and gives recommendations about parameter settings.

5. General discussion

Using the Rosenbrock function in 2 dimensions we graphically analyze the effects of the estimation/selection steps. Table 4 shows several typical generations of the algorithm. In Table 4(a) the initial large sample $\hat{X}$ is shown with small dots (see Step 1, Algorithm 3), using the Maximin algorithm we select the initial population which is shown with big dots in Table 4(a) and with small dots in Table 4(b). As can be seen, the initial population are individuals with high diversity, well spread over the whole search space.

After evaluation the fittest individuals are selected according to the truncation selection, the selected set is shown with big dots in Table 4(b). Table 4(c) shows the evolution of the population after 10 generations, we generate a large sample (small dots) and then select some of them to repopulate (big dots with a small dot inside). The big dots that have a small dot inside are new individuals selected to repopulate. The big dots without a small dot inside are individuals that have been preserved from the last generation. In Table 4(d) we can observe that the selected set (big dots with a small dot inside) adequately recovers the structure of the function landscape, this selected set will be preserved to the next generation, preserving as well, the information about the function landscape. Table 4(e) shows the population after 100 generations, the algorithm almost converges. At this resolution the population seems collapsed, but if we take a look at Table 4(f) which is a close up, we can see that the selected set (big dots) and the population (small dots) are well distributed in the promising area. These plots show how the EDA-SRP works, the truncation method promotes the convergence and reduction of the search region, the Maximin promotes the diversity in the population and selected set. The plots in this section show that our claims of generating high diversity solutions, and preserving informative solutions are well supported. Table 4(d), shows that the selected set (big dots) are individuals which contain sufficient information to determine the adequate structure, density and position of an adequate search distribution. It is clear that if the Normal distribution could better follow the selected set in Table 4(d) the algorithm becomes more efficient. This last statement is supported by the results, the EDA-SRP approximates the optimum with an error less than 1e−10 using 87341.3 ± 968.36, $2.092e6 \pm 1.5e5$ and 280937.4 ± 19656.7 function evaluations, for the Sphere, Rosenbrock and Ackley functions respectively. The sphere is the problem which needs the minimum number of function evaluations for the EDA-SRP. That is to say, if the objective function is a single mode function with an ellipsoidal shape (just like the Normal distribution), the optimum is efficiently approached. The problems arise when the structure of the function is different from the structure of the Normal distribution.

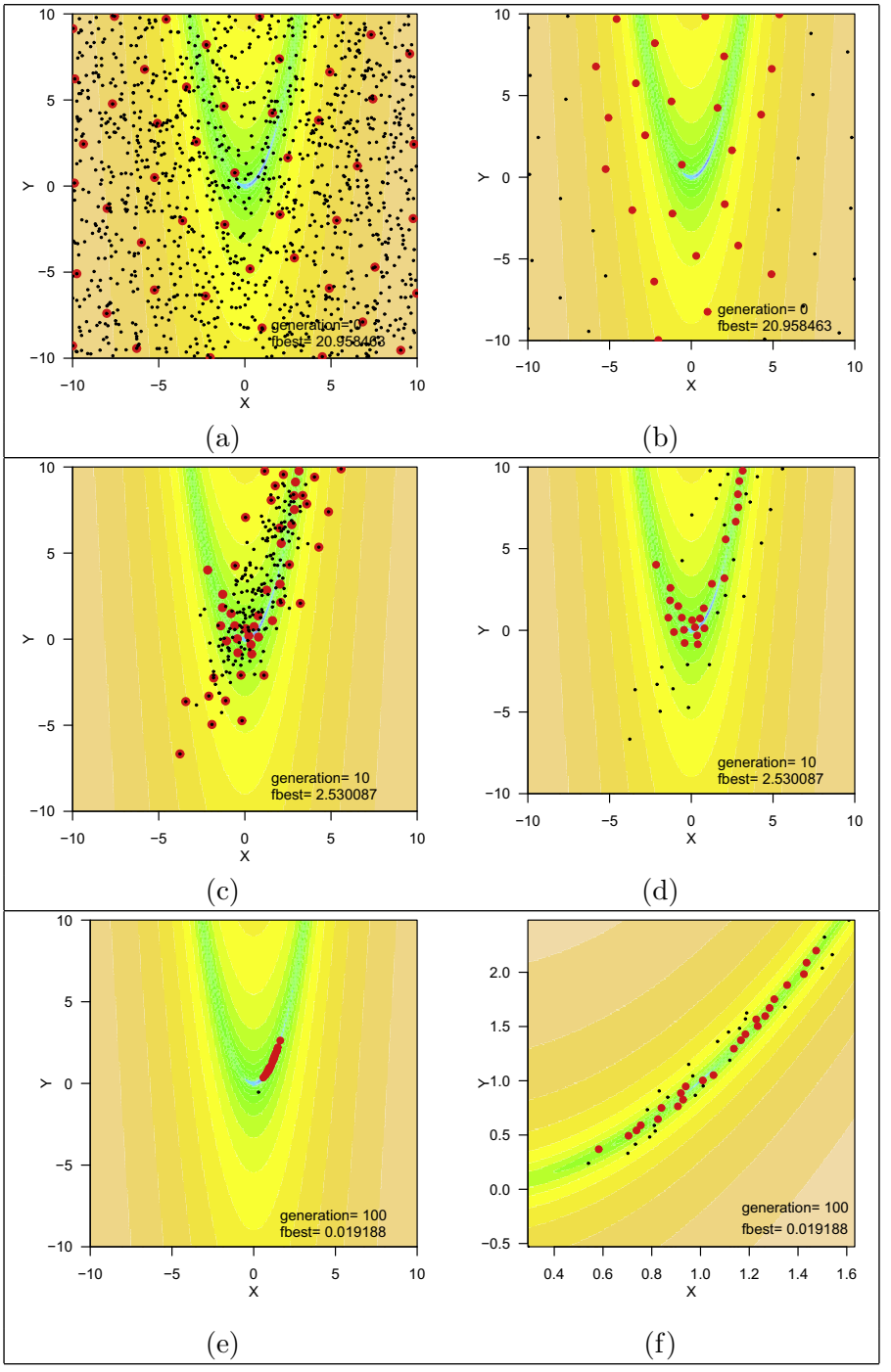
5.1. Parameter settings

The user-given parameters for the EDA-SRP are the population size and the resampling rate n_{rs}. Using the Rosenbrock function, Fig. 1 shows the number of function evaluations needed to reach an objective value of 1e−10, for $n_{rs} = 3$ and $n_{rs} = 4$. The minimum function evaluations are obtained for a population size of 160.

As can be seen, this parameter does not have an important impact for this problem, in our experiments it is set to 3, except for the Schwefel f_8. The reason is that n_{rs} delivers an objective mean of -10103.34 ± 551.6, which is worst than the one reported in Table 3. The explanation is that an small n_{rs} parameter promotes faster convergence, this is supported by the smaller standard deviation for the Schwefel f_8 for $n_{rs} = 3$. Our experiments suggest that n_{rs} smaller than 3 is only useful for functions such as the sphere, which has a similar structure as the Normal. On the other hand, a n_{rs} larger than 6 has no significant impact on the results, but it has impact on the computational cost. According to the boxplots in Fig. 1 we suggest an $n_{rs} = 3$, in general, and $n_{rs} = 4$ for multi-modal hard functions.

The second conclusion elucidated from Fig. 1, is that there is an *optimal* population size for the *EDA-SRP*, This second conclusion is supported by Fig. 2, the best objective function value is plotted versus the population size, the objective value axis is in log-scale. As can be observed when the population size is large enough the algorithm can find close approximations to the optimum, but the computational cost increases. The algorithm also is stopped if the number of evaluations reaches 1e5, hence some runs with the largest population sizes do not converge. Fig. 2, shows that the most effective population size is

Table 4
Several generations of a typical run of EDA-SRP with the Rosenbrock function.



between 160 and 240. Our recommendation is to set a small population size and increase it until the performance does not change. For the sake of completeness the success rate (ratio of successful runs) is $\{0.533, 0.7, 1, 0.97, 0.87, 0.97\}$ for the population sizes of $\{80, 160, 240, 320, 480, 960\}$ for $n_{rs} = 3$ and 1e5 function evaluations. And, $\{0.370.5710.930.870.77\}$, for $n_{rs} = 4$, the smaller success rate for the last runs can be explained by slower convergence when the re-sampling rate is increased. Finally, the success performance, defined as the mean of function evaluations of successful runs, divided by the suc-

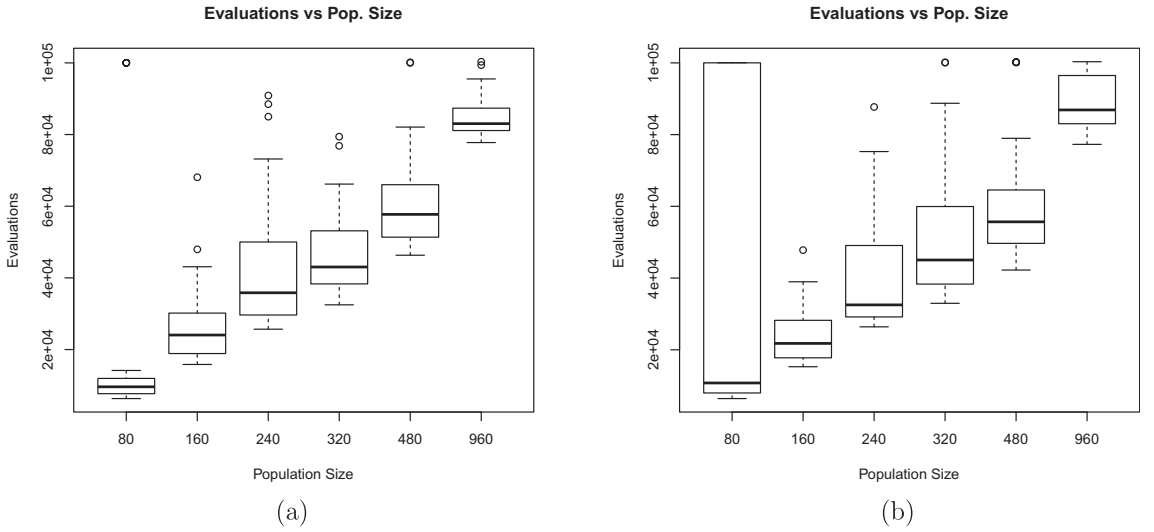


Fig. 1. Boxplots for the number of function evaluations versus the population size for 30 independent runs of the EDA-SRP with the 5-D Rosenbrock function. Only successful runs are used (objective value $\leqslant$1e−10). (a) Resampling rate $n_{rs} = 3$, (b)$n_{rs} = 4$.

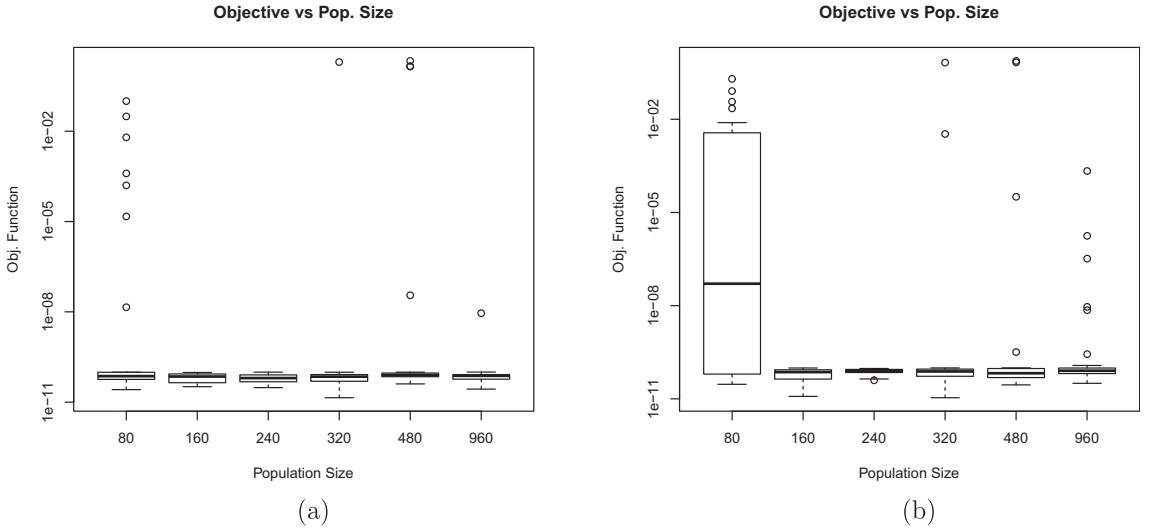


Fig. 2. Boxplots for the best function value versus the population size for 30 independent runs of the EDA-SRP with the 5-D Rosenbrock function. All runs are used. (a) Resampling rate $n_{rs} = 3$, (b)$n_{rs} = 4$.

cess rate is $\{37153.70, 38502.48, 43636.77, 48807.01, 71299.90, 87856.55\}$ and $\{142341.60, 42728.43, 41360.83, 55620.69, 71064.81, 116295.65\}$ for $n_{rs} = 3$ and $n_{rs} = 4$. Notice how similar they are for the population size of 160 and 240, this confirms that the performance is robust to the n_{rs} parameter, and that an adequate population size can be found relatively easily.

5.2. Individual contribution of the enhancement techniques

The goal of this section is twofold: Firstly we present a comparison among the proposal, a standard EDA, and the standard EDA enhanced with the techniques just proposed but once at a time. The purposes of this subsection are: to known what they can be used for, the individual contribution of each of them, and if it is possible to use only one of them which provides an specific behavior according to the problem at hand. Secondly, we show the performance of our proposal with other well known objective functions in order to provide a better understanding of it. The standard EDA used as basis for this comparison works as follows:

Table 5
Test problems used for comparing the effects and impacts in the performance of the different enhancement techniques proposed in this work.

Name	Definition	Domain
Rosenbrock	$\sum_{i=1}^{n-1}\left(100(x_i - x_{i+1}^2)^2 + (x_i - 1)^2\right)$	$x \in \{-10, 5\}^n$
Ackley	$f(x) = -20 \cdot \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	$x \in \{-32.768, 16.384\}^n$
Griewank	$f(x) = \frac{\sum_{i=1}^{n} x_i^2}{4000} - \prod_i^n \cos(x_i/\sqrt{i}) + 1$	$x \in \{-600, 600\}^n$
Ellipsoid	$f(x) = \sum_i^n 10^{6(i-1)/(n-1)} x_i^2$	$x \in \{-10, 5\}^n$
Cigar	$f(x) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$	$x \in \{-10, 5\}^n$
Cigar Tablet	$f(x) = x_1^2 + 10^4 \sum_{i=2}^{n-1} x_i^2 + 10^8 x_n^2$	$x \in \{-10, 5\}^n$
Two Axes	$f(x) = 10^6 \sum_{i=1}^{n/2} + \sum_{i=n/2}^{n} x^i$	$x \in \{-10, 5\}^n$
Different powers	$f(x) = \sum_i^n \lvert x_i\rvert^{2+10\frac{i-1}{n-1}}$	$x \in \{-10, 5\}^n$

1. Generate an initial population of n_{pop} individuals by means of the uniform distribution in the search space.
2. Evaluate the population.
3. Select the best $n_{pop}/2$ individuals in the population.
4. Estimate a Normal multivariate distribution by using the maximum likelihood estimator of the mean and covariance matrix given the selected set.
5. Sample the Normal model to get $n_{pop}/2$ new individuals, evaluate them and combine them with the selected set to obtain the new population.
6. Repeat from step 3.

The aim of this comparison is not to decide which algorithm performs the best, but to understand the effect contributed by each enhancement. We do not tune the parameters to avoid an unfair comparison, which could lead us to erroneous conclusions, due to the indistinguishable performance of the different versions of the algorithms. Additionally, the **stopping criteria** are: the covariance matrix norm must be less than 10^{-50}, the covariance matrix must be numerically positive definite, and the number of evaluations must be less than 50×10^3. We use this values in order to show the performance of the different techniques. If more function evaluations are used, many of the algorithms does seem to perform similar, hence this values help to show the actual performance. The algorithms contrasted in this subsection are the following.

- EDA-Std. The standard EDA implemented as mentioned above.
- EDA-Init. Similar to the EDA-Stp, but modified in the initial population, which is generated according to the proposal in the EDA-SRP to improve the initial diversity and exploration.
- EDA-Rpop. The difference with the EDA-Std is the repopulation step. A large set of candidate solutions is generated, then some of them are rejected as in the EDA-SRP using the Maximin and the nearest neighbor weight. Neither the initialization, nor the weighted estimators, nor the truncation selection of the proposal are used.
- EDA-West. Is the EDA-Std but using the weighted estimators.
- EDA-Trunc. Is the EDA-Std but using the truncation method.
- EDA-SRP. Is the EDA with all the enhancements proposed in this article.

The objective functions used in this tests are show in Table 5. They are well known in global optimization, and most of them have been used to test the performance of Normal Multivariate EDAs [23].

The results are presented using violin plots in Fig. 3. The violin plots represent the density of the best solutions found in 15 independent runs. They deliver an accurate idea about the variance of the solutions and the actual positions. Violin plots can help to compare the density of the best solutions found by different algorithms instead of a single value such as the mean, additionally the mean value can be also graphically compared (the white dot in the middle of the violin). This kind of comparison seems quite fair or at least more complete than a punctual one. For sake of completeness we present the mean and standard deviation of this comparison in Table 6.

5.3. Discussion about comparison of different techniques

Looking at Fig. 3 and Table 6 we can notice that the Ackley function is solved by all the algorithms but the EDA-Rpop which has the component of the selective repopulation. Considering that the EDA-SRP does not converges as close as the other algorithms, we can conclude that the convergence of the algorithms with repopulation is slower than the others. Additionally, we can infer that the EDA-SRP has an extra component which leads it to a faster convergence than the EDA-Rpop. The functions: Cigar, Cigar Tablet, Ellipsoid and Two Axes present a similar pattern, the best performed algorithm in these

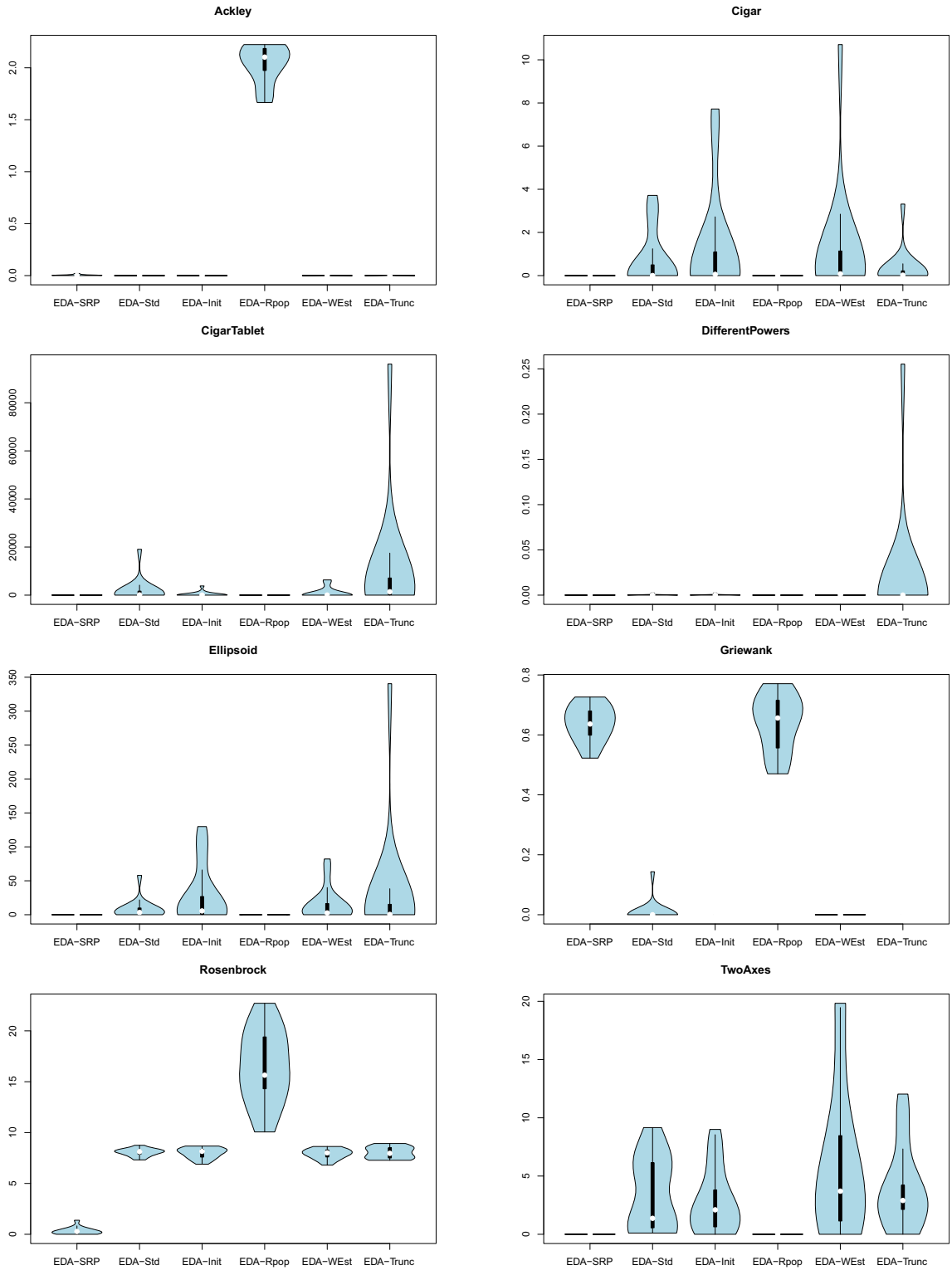


Fig. 3. Comparison of different techniques used in this paper to improve the EDA performance.

Table 6
Mean and (Standard deviation), for different test problems, and algorithms modified with the different techniques proposed in this article.

Problem	EDA-SRP	EDA-Std	EDA-Init	EDA-Rpop	EDA-WEst	EDA-Trunc
Rosenbrock	0.3187	8.06	7.985	16.42	7.907	8.028
	(0.3612)	(0.3659)	(0.5316)	(3.622)	(0.5164)	(0.5799)
Ackley	0.004263	6.809e−16	9.178e−16	2.034	0.0002638	0.0006226
	(0.005744)	(9.173e−16)	(1.25e−15)	(0.1863)	(0.001022)	(0.002411)
Griewank	0.6388	0.009561	0	0.6377	4.433e−15	0
	(0.0611)	(0.03703)	(0)	(0.09588)	(1.717e−14)	(0)
Ellipsoid	2.78e−06	8.345	27.69	1.308e−18	15.17	29.69
	(1.072e−05)	(14.54)	(43.24)	(6.002e−19)	(24.35)	(86.87)
Cigar	3.535e−27	0.6997	1.275	3.049e−20	1.186	0.3359
	(1.369e−26)	(1.3)	(2.507)	(2.22e−20)	(2.756)	(0.8479)
Cigar Tablet	0.05326	1993	422.3	2.114e−06	982.6	10281)
	(0.1787)	(4889)	(1027)	(6.92e−07)	(2144)	(24509)
Two Axes	4.075e−08	3.364	2.795	2.871e−18	5.942	3.933
	(1.578e−07)	(3.182)	(2.816)	(2.133e−18)	(6.476)	(3.589)
Different powers	4.716e−27	0.000161	0.0001599	4.297e−19	9.727e−07	0.01702
	(1.824e−26)	(0.0006234)	(0.0006192)	(1.647e−18)	(3.738e−06)	(0.06591)

cases are the EDA-SRP and EDA-Rpop. A common characteristic of these test functions is that the variables have a quite different weight in the objective function, the EDA-Std reduces the variance in all directions equally, as consequence it suffers of premature convergence because it favors the highly weighted variables. The different powers function shows that the truncation method in EDA-Trunc could favor the premature convergence if it is not used together with a technique which favors the exploration. On the other hand, Griewank function in Table 6 shows that the best performed algorithm is the one equipped with the truncation method, and the worst performed algorithm are the ones with the repopulation technique. Summarizing, using the information collected from all the algorithms we can conclude that: the truncation method favors convergence or exploitation of the information, while the repopulation method favors exploration. The weighting procedure equips the EDA with a more efficient estimation-sampling step, in the sense that it consistently shows a lower objective function than the other algorithms, hence, with less function evaluations it estimates better than the others the optimum, note that the violin plot of EDA-West consistently is smaller (low variance) and lower (better objective) than the others. Finally, Rosenbrock function shows that the techniques proposed in this work could result in a balance between exploration and exploitation when used together.

6. Conclusions

The EDA proposed in this work intends to circumvent important issues of classical Normal EDAs, such as diversity preservation and preservation of the necessary information about the fitness landscape. Notice that this proposal does not guarantee that the population or the selected set actually are Normally distributed, because of the selective re-population and preservation of the fittest individuals. Instead, we intend to preserve solutions which indicate the function landscape in the most promising regions, additionally, we intend to sample candidate solutions as diverse as possible. These concepts could be different from the standard estimating and sampling process, which uses parametric distributions to fit the selected set and sample similar solutions to those already known. As a possible negative effect of the EDA-SRP way of working, is the loss of the direct relationship between the search distribution and the population. On the other hand, an advantage is that the Normal distribution is only an engine to generate random candidate solutions, thus we expect similar promising results if other sampling engines are used, or if we apply the same idea in other domains. This means that the ideas in this article can be extended to other kinds of problems and/or algorithms. The EDA-SRP is competitive with similar EDAs and classical approaches as is shown in the experiments performed. We have shown that the obtained results are even better than the results obtained by more complex models based on Normal mixtures. In addition, the EDA-SRP uses a small population size, which indicates that it uses the information better than other approaches, and generates more informative candidate solutions. A small sized population also impacts in the memory and the computational cost used by the algorithm (the Maximin is quadratic with the population size). The experiments conducted suggest that the truncation selection promotes convergence while the Maximin selection leads to a smooth and slow convergence as a payoff of maintaining individuals with high diversity.

Notice that our proposal inserts techniques that are not dependent on a Normal Multivariate model, and in some of the cases, neither to a continuous search space. Hence, the Maximin ranking, the truncation method, the weighted estimators and preselection of candidate solutions to be evaluated, as well as a diverse initialization are ideas that can be used in other population based algorithms.

A conceptual conclusion is the need to look for search engines which intensively sample the most promising regions, while sampling candidate solutions with high diversity. Neither maximum likelihood (ML) estimation, nor some variance

scaling techniques seem to follow this paradigm. Researchers have advised that Normal EDAs must use the covariance matrix which improves the results, instead of the one which fits the selected set [12,8,10,11,23,24]. They also have avoided evaluating all the individuals, by discriminating the non-promising ones, according to predictions delivered by local models [20]. In this article we use the mentioned strategies and add two more: (1) we avoid evaluating individuals that are in regions that have been already evaluated by using the Maximin algorithm and, (2) we preserve valuable information about the objective function structure.

The general conclusion is to generate and to preserve informative individuals. Such informative individuals must provide information that no other individual in the population provides, locally each of them must be isolated, if not, such information is redundant. Furthermore, the information provided by the individuals must be valuable for the search process, that is to say, it must help to elucidate the objective function landscape and, eventually, the optimum position. The EDA-SRP is a proposal that agrees with this conclusion, but obviously, other strategies could be followed to perform similar improvements.

Future work will contemplate a discrete version of the EDA-SRP, the use of Normal mixtures and local meta-models [20] instead of the simple ranking scheme, to reduce the number of function evaluations.

Algorithm 1: Maximin algorithm to rank solutions according their diversity.

Input:
R Set of reference points, where $r_j \in R, j = 1..nr$;
X Points to be ranked, where $x_i \in X, i = 1..nx$;
$nvar$ Number of dimensions.

```
for i = 1..nx do
    d_i = min_j(euclidian_distance(x_i, r_j));
for i = 1..nx do
    i^max = k such that d_k ≥ d_i  ∀ i ∈ {1, 2, .., nx};
    Rank_{i^max} = i;
    d_{i^max} = −1;
    for j = 1..nx do
        if d_j > −0.1 then
            d_j = min(d_j, euclidian_distance(x_{i^max}, x_j))
```

Output:
$Rank$ A rank attached to each point in the X set;

Algorithm 2: Truncation method to ensure an increasing mean (bounded by an increasing sequence) of the objective function and convergence to the elite individual. Maximization case.

Input:
F vector of objective function values of size n;
θ^t a threshold, for the first selection $\theta^0 = \min(F)$;

```
I ← sort(F,decreasing,return_index);
ε ← 1e − 14(max(|F_{I_0}|, |F_{I_n}|, |F_0 − F_n|));
k ← n/2;
while k > 0.05n  &  F_{I_k} < (θ^t + ε) do
    k ← k − 1
θ^{t+1} ← F_{I_k};
```

Output:
$I_{1:k}$ vector of indexes of selected individuals of size k;
θ^{t+1} threshold for the next selection;

Algorithm 3: Normal multivariate EDA with selective repopulation.

$\hat{X} \leftarrow \text{uniform}(6n_{rs} \cdot n_{pop}, x_{inf}, x_{sup})$;
$R \leftarrow \text{reference\_points}(\hat{X}, x_{inf}, x_{sup})$;
$n_S \leftarrow \text{sizeof}(R)$;
$X \leftarrow \text{maximin\_selection}(\hat{X}, R)$;
$F \leftarrow \text{evaluation}(X)$;
$I \leftarrow \text{truncation\_selection}(F)$;
$W \leftarrow \text{weight\_computation}(F[I])$;
$R \leftarrow X[I]$;
$F_R \leftarrow F[I]$;
Compute μ and Σ using W;
while $|\Sigma| > \epsilon_{tol}$ & $max_{eval} < n_{eval}$ **do**
  $\bar{X} \leftarrow \text{Normal}(n_{rs} \cdot n_{pop}, \Sigma, \mu, x_{inf}, x_{sup})$;
  $Rank^{maximin} \leftarrow \text{maximin\_ranking}(\hat{X}, R)$;
  //Computing a rank to select the individuals to be evaluated
  for $j = 1..n_{pop}$ **do**
    $k = \text{nearest\_neighbor}(R, \bar{x}_j)$;
    $Rank_j^* = \frac{(w_k)}{Rank_j^{maximin}}$
  //Copying the first $(n_{pop} - n_S)$ indices to J
  $J \leftarrow \text{sort}(Rank^*, \text{decreasing}, \text{return\_index})[1 : (n_{pop} - n_S)]$;
  $\bar{F} \leftarrow \text{evaluation}(\hat{X}[J])$;
  //Repopulating
  $X \leftarrow R \cup \bar{X}[J]$;
  $F \leftarrow F_R \cup \bar{F}$;
  $I \leftarrow \text{truncation\_selection}(F)$;
  $x_{best} \leftarrow X[I[0]]$;
  $R \leftarrow X[I]$;
  $F_R \leftarrow F[I]$;
  $n_S \leftarrow \text{sizeof}(R)$;
  $W \leftarrow \text{weight\_computation}(F)$;
  Compute μ and Σ;
Output:
x_{best} Best optimum approximation.

References

[1] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic Publishers, Norwell, MA, USA, 2001.
[2] H. Mühlenbein, G. Paa, From recombination of genes to the estimation of distributions I. Binary parameters, in: H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature—PPSN IV, Lecture Notes in Computer Science, vol. 1141, Springer, Berlin/Heidelberg, 1996, pp. 178–187, http://dx.doi.org/10.1007/3-540-61723-X982.
[3] P.A.N. Bosman, D. Thierens, Expanding from discrete to continuous estimation of distribution algorithms: the IDEA, in: PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, London, UK, 2000, pp. 767–776.
[4] P.A. Bosman, J. Grahl, D. Thierens, Enhancing the performance of maximum—likelihood gaussian edas using anticipated mean shift, in: Proceedings of the 10th International Conference on Parallel Problem Solving from Nature: PPSN X, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 133–143.
[5] J.L. Shapiro, Diversity loss in general estimation of distribution algorithms, in: Proceedings of the 9th International Conference on Parallel Problem Solving from Nature, PPSN'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 92–101.
[6] W. Dong, X. Yao, Unified eigen analysis on multivariate gaussian based estimation of distribution algorithms, Inf. Sci. 178 (2008) 3000–3023.
[7] Q. Zhang, H. Mühlenbein, On the convergence of a class of estimation of distribution algorithms, IEEE Trans. Evol. Comput. 8 (2004) 127–136.
[8] P.A. Bosman, J. Grahl, Matching inductive search bias and problem structure in continuous estimation-of-distribution algorithms, Eur. J. Oper. Res. 185 (2008) 1246–1264.
[9] B. Yuan, M. Gallagher, On the importance of diversity maintenance in estimation of distribution algorithms, in: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, GECCO'05, ACM, New York, NY, USA, 2005, pp. 719–726.
[10] Y. Cai, X. Sun, H. Xu, P. Jia, Cross entropy and adaptive variance scaling in continuous EDA, in: Proceedings of the 9th annual conference on Genetic and evolutionary computation, GECCO'07, ACM, New York, NY, USA, 2007, pp. 609–616.
[11] J. Grahl, P.A.N. Bosman, F. Rothlauf, The correlation-triggered adaptive variance scaling IDEA, in: GECCO'06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, ACM Press, 2006, pp. 397–404.
[12] O. Kramer, F. Gieseke, Variance scaling for EDAs revisited, in: Proceedings of the 34th Annual German conference on Advances in artificial intelligence, KI'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 169–178.

[13] D. Cho, B. Zhang, B. Labratory, Evolutionary continuous optimization by distribution estimation with variational bayesian independent component analyzers mixture model, in: Proceedings of Parallel Problem Solving from Nature VIII, Lecture Notes in Computer Science, volume 3242, Springer, 2004, pp. 212–221.
[14] P.A. Bosman, D. Thierens, Advancing continuous IDEAs with mixture distributions and factorization selection metrics, in: Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference GECCO2001, Morgan Kaufmann, 2001, pp. 208–212.
[15] J. Ocenasek, J. Schwarz, Estimation of distribution algorithm for mixed continuous–discrete optimization problems, in: 2nd Euro-International Symposium on Computational Intelligence, IOS Press, 2002, pp. 227–232.
[16] M. Wagner, A. Auger, M. Schoenauer, EEDA: a new robust estimation of distribution algorithms, Research Report RR-5190, INRIA, 2004.
[17] N. Hansen, A. Ostermeier, Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation, in: Proceedings of IEEE International Conference on Evolutionary Computation, 1996, pp. 312–317.
[18] S.I. Valdez, S. Botello, A. Hernández, Uniformly distributed pareto fronts through the maximin selection algorithm, Int. J. Artif. Intell. Tools (2009) 355–362.
[19] F. Teytaud, O. Teytaud, Why one must use reweighting in estimation of distribution algorithms, in: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO'09, ACM, New York, NY, USA, 2009, pp. 453–460.
[20] Z. Bouzarkouna, A. Auger, D.Y. Ding, Investigating the local-meta-model cma-es for large population sizes, in: EvoApplications (1), Lecture Notes in Computer Science, vol. 6024, Springer, 2010, pp. 402–411.
[21] W. Dong, X. Yao, NichingEDA: utilizing the diversity inside a population of EDAs for continuous optimization, in: IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence), pp. 1260–1267.
[22] Q. Lu, X. Yao, Clustering and learning gaussian distribution for continuous optimization, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 35 (2005) 195–204.
[23] P.A.N. Bosman, J. Grahl, F. Rothlauf, SDR: a better trigger for adaptive variance scaling in normal EDAs, in: GECCO'07: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, ACM, 2007, pp. 516–522.
[24] J. Grahl, P.A.N. Bosman, S. Minner, Convergence phases, variance trajectories, and runtime analysis of continuos EDAs, in: GECCO'07: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, ACM, 2007, pp. 516–522.