A Boltzmann Multivariate Estimation of Distribution Algorithm for Continuous Optimization

Ignacio Segovia[1], Second Author Name[1] and Third Author Name[2]
[1]*Center for Research in Mathematics A.C., Guanajuato, Mexico. C.P. 36240*
ijsegoviad, s_author}@cimat.mx

Keywords: Boltzmann distribution, Estimation of Distribution Algorithm, Optimization.

Abstract: This article introduces an approach for continuous optimization using an Estimation of Distribution Algorithm (EDA), based on the Bolztmann distribution. When using the objective function as energy function, the Boltzmann function favors the most promising regions, making the probability exponentially proportional to the objective function. Using the Botlzmann distribution directly for sampling is not possible because it requires to compute the objective function values in the complete search space. In this work we propose an approximation to the Boltzmann function by a multivariate normal distribution. Formulae for computing the mean and covariance matrix is derived by minimizing the Kulback-Leibler divergence. The proposal is competitive and often superior to similar algorithms as it is shown by statistical results reported in this paper.

1 INTRODUCTION

Estimation of Distribution Algorithms (Müehlenbein[1] et al., 1996) are optimization methods based on estimating and sampling a probability distribution. The aim is to favor the most promising regions assigning them the highest probability values. Actually, the most promising regions are unknown and have to be discovered during the optimization process. The main goal of the EDA is to pose the probability mass around the optima, the strategy, without loss of generality for a maximization process, is to reinforce the sampling in regions with the maximum objective function or fitness function values, and disregard the regions with the minimum values. The most common scheme for continuous optimization using EDAs, is to use a multivariate or univariate Normal distribution (Larraaga, 2002; Larrañaga et al., 2000; Dong and Yao, 2008), the Normal parameters are estimated by using maximum likelihood estimators (MLEs) over the selected set, which is determined by the truncation method, usually the half of the population with the worst objective value is truncated. Nevertheless these approaches have shown a competitive performance, some evident issues can be noticed in the strategy just mentioned:

- The truncation selection hides the fitness landscape assigning to the selected individuals the same importance in the parameter estimation, in consequence, the search distribution parameters are estimated as if the regions represented by the selected set were equally good.
- It is a well known issue that the variance in estimation of distribution algorithms often is less than required, hence, the MLE variance estimator is not the most adequate for searching the optimum (Shapiro, 2006; Grahl et al., 2007).

The Botlzmann distribution has been largely used in optimization, in the Estimation of Distribution Algorithms (EDAs) context researchers have proposed different approaches such as the BEDA (Mhlenbein, 2012; Mahnig and Muhlenbein, 2001; Mühlenbein et al., 1999; Mühlenbein et al., 1999), which is a general framework for Boltzmann based estimation of distribution algorithms, where practical EDAs have been derived from, as the FDA (Muhlenbein and Mahnig, 1999), in the same line is the Yun peng et al. proposal (Yunpeng et al., 2006) and Valdez et al. proposal. (Valdez et al., 2013). The unifying characteristic of this approaches is that, the intend to equip stochastic search algorithm with an engine which favors the most promising regions. The better the objective function is in a region the most intensive the sampling must be. The Boltzmann distribution is used to achieve such purpose.

The Gibbs or Boltzmann distribution of an fitness function g(x) is defined by:

$$p(x) := \int_X \frac{exp(\beta g(x))}{Z} \tag{1}$$

As can be notice in Equation 1 the objective function is used as energy function directly. In practical approaches the Boltzmann distribution can not be used directly for sampling because it is necessary to known the objective function in the whole domain. That is the reason of using a parametric distribution which parameters are computed to minimize a distance measure between the parametric distribution and the Boltzmann distribution. Several proposals had explored this main idea: to approximate a Boltzmann distribution to a parametric distribution by minimizing a distance measure, by instance, the Kullback-Leibler divergence (Ochoa, 2010; Yunpeng et al., 2006; Valdez et al., 2013).

Nevertheless, most of them are competitive, there are some remarkable challenges when designing EDAs based on the Boltzmann distribution, such as the listed below:

- To choose an adequate β parameter in Equation 1. Usually β depends on the time or is dynamic during the optimization process, the function which controls the β updating each generation is called *the annealing schedule*. The annealing schedule can be used to manage the exploration and convergence of the algorithm.
- To derive robust formulae to different β, problems, and populations. Some approaches (Hu et al., 2012; Yunpeng et al., 2006) had derived formulae for estimating parameters of a distribution which approximate the Boltzmann, by weighting the population or selected set by exponential functions, similar to Equation 1. Even though competitive results are obtained, the proposals often suffer from premature convergence, because the exponential function drastically leads the probability mass to suboptimal positions, this behavior can be avoided by manipulating the β value, but it is not simple to determine how to do it, as second option is to obtained formulae which do not impact so drastically the estimators.
- The last two issues also are related with the mentioned variance reduction which is a common issue in EDAs (Shapiro, 2006).

According to the challenges just mentioned, our proposal intends to tackle all of them. We present a novel algorithm, based on the approximation of the Boltzmann function by a Normal multivariate distribution, which introduces the following features:

- Two proposals of annealing schedules to update the β value.
- Formulae which is robust or a least is not impacted as drastically as the exponential function of changes int the population or the β value.
- Our proposals of annealing schedules tackle the variance reduction problem, hence it is a mechanism to avoid a premature convergence of the algorithm.

The organization of the paper is as follows: Section 2 presents the formulae derivation for computing the parameters of the Normal multivariate distribution, Section 3 introduces the Boltzmann Estimation of Multivariate Normal Algorithm (BEMNA), as well as two annealing schedules used in it. Section 4 presents statistical results on well known test functions, and finally 5 presents the main conclusions of this work.

2 APPROXIMATING THE BOLTZMANN DISTRIBUTION BY THE NORMAL MULTIVARIATE DISTRIBUTION

In this section we introduce formulae to estimate the mean and standard deviation, it is to say $\vec{\mu}_*$ and Σ_*, of a Normal multivariate density which approximate the multivariate Boltzmann P_x density, given a set of samples $\vec{x}^{(1)}, \vec{x}^{(2)}, ..., \vec{x}^{(N)}$ which are observations of a random vector $\vec{X}$. Let $\vec{X}$ be a random vector such that $\vec{X} \sim Q_x$, where $Q_x = Q(x, \mu, \Sigma)$ is the multivariate Normal density as shown in Equation 2. The corresponding Boltzmann density is in Equation 3.

$$Q(_x) = \frac{1}{\sqrt{(2\pi)^d \, |\Sigma|}} \exp\left\{ -\frac{1}{2} (\vec{x} - \vec{\mu})^t \Sigma^{-1} (\vec{x} - \vec{\mu}) \right\}. \tag{2}$$

$$P_x = \frac{\exp(\beta g(\vec{x}))}{Z}, \tag{3}$$

The procedure for finding the parameters of Q_x which best approximate P_x consist in minimizing a measure of dissimilarity between density functions. Similar to previous works (Yunpeng et al., 2006) (Valdez et al., 2013), we use the Kullback-Leibler Divergence presented in Equation 4 $D_{KL}(Q_x||P_x)$, for short written as K_{QP}.

$$K_{QP} = \int Q_x \log \frac{Q_x}{P_x} \, d\vec{x}. \tag{4}$$

The minimization of K_{QP} for finding the optimal parameters $[\vec{\mu}_*, \Sigma_*]$ can be stated as shown in Equation 5.

$$[\vec{\mu}, \Sigma] = \arg\min \{K_{QP}\} \tag{5}$$

We ca rewritting K_{QP} as shown in Equation 6.

$$\begin{aligned} K_{QP} &= \int Q_x \log Q_x \, d\vec{x} - \int Q_x \log P_x \, d\vec{x} \\ &= -H(Q_x) - \int Q_x \log P_x \, d\vec{x} \\ &= -\frac{1}{2}\log((2\pi e)^d \, |\Sigma|) - \int Q_x \log P_x \, d\vec{x}. \end{aligned} \tag{6}$$

Where the term $H(Q_x)$ means the entropy of the multivariate Normal density (Cover and Thomas, 2006). In order to find the parameters which minimize the Kulback-Leibler Divergence, we derive respect to each parameter as shown in Equations (7) and (8).

$$\begin{aligned} \frac{\delta K_{QP}}{\delta \vec{\mu}} &= -\int \frac{\delta Q_x}{\delta \vec{\mu}} \log P_x \, d\vec{x} \\ &= -\int Q_x [\Sigma^{-1}(\vec{x} - \vec{\mu})] \log P_x \, d\vec{x} \end{aligned} \tag{7}$$

$$\begin{aligned} \frac{\delta K_{QP}}{\delta \Sigma} &= -\frac{1}{2}\frac{\delta \log(|\Sigma|)}{\delta \Sigma} - \int \frac{\delta Q_x}{\delta \Sigma} \log P_x \, d\vec{x} \\ &= -\frac{1}{2}\int Q_x [\Sigma^{-1} \, (\vec{x} - \vec{\mu}) \, (\vec{x} - \vec{\mu})^t \, \Sigma^{-1}] \log P_x \, d\vec{x} \\ &\quad + \frac{1}{2}\int Q_x \Sigma^{-1} \log P_x \, d\vec{x} - \frac{1}{2}\Sigma^{-1} \end{aligned} \tag{8}$$

The optimal estimates for the mean and covariance matrix are obtained by making the derivatives equal to 0, as in Equations 7 and 8, and solving for $\vec{\mu}$ and Σ respectively.

$$\begin{aligned} 0 &= \frac{\delta K_{QP}}{\delta \vec{\mu}} \\ 0 &= \vec{\mu} \int Q_x \log P_x \, d\vec{x} - \int Q_x \vec{x} \log P_x \, d\vec{x} \\ 0 &= \vec{\mu} \beta E_Q[g(\vec{X})] - \vec{\mu} \log Z - E[g(\vec{X})\vec{X}]\beta + E[\vec{X}] \log Z \\ \vec{\mu} &= \frac{E_Q[g(\vec{X})\vec{X}]}{E_Q[g(\vec{X})]} \end{aligned} \tag{9}$$

$$\begin{aligned} 0 &= \frac{\delta K_{QP}}{\delta \Sigma} \\ 0 &= \int Q_x \, (\vec{x} - \vec{\mu}) \, (\vec{x} - \vec{\mu})^t \log P_x \, d\vec{x} \\ &\quad - \int Q_x \log P_x \, d\vec{x} \Sigma + \Sigma \\ \Sigma &= (E_Q[g(\vec{X})] - 1/\beta)^{-1} E_Q[g(\vec{X})(\vec{X} - \vec{\mu})(\vec{X} - \vec{\mu})^t] \end{aligned} \tag{10}$$

The following equivalences where used to get Equations 9 and 10:

- $\log P_x = \beta g(\vec{x}) - \log Z$,
- $\int Q_x \vec{x} \, d\vec{x} = E_Q[\vec{X}], \quad \int Q_x (\vec{x} - \vec{\mu})(\vec{x} - \vec{\mu})^t \, d\vec{x} = E_Q[(\vec{X} - \vec{\mu})(\vec{X} - \vec{\mu})^t]$.
- As $\vec{X} \sim Q_x$ then $E_Q[\vec{X}] = \vec{\mu}$, and $E_Q[(\vec{X} - \vec{\mu})(\vec{X} - \vec{\mu})^t] = \Sigma$.

Finally, for estimating the parameters using the observations $\vec{x}^{(1)}, \vec{x}^{(2)}, ..., \vec{x}^{(N)}$ of the random variable $\vec{X}$, we use a numerical stochastic approximation by using the Monte Carlo method, as it is shown in Equations 11 and 12. These two equations will be used int the algorithm for estimating the parameters of the search distribution.

$$\vec{\mu}_* = \frac{\sum_{i=1}^N g(\vec{x}^{(i)}) \vec{x}^{(i)}}{\sum_{i=1}^N g(\vec{x}^{(i)})} \tag{11}$$

$$\Sigma_* = r_e \cdot \sum_{i=1}^N g(\vec{x}^{(i)})(\vec{x}^{(i)} - \vec{\mu})(\vec{x}^{(i)} - \vec{\mu})^t \tag{12}$$

where

$$r_e = \left(\sum_{i=1}^N g(\vec{x}^{(i)}) - \frac{N}{\beta} \right)^{-1} \tag{13}$$

2.1 A note about the derived formulae

In Equations 11 and 12 the estimators uses weights defined by $\frac{\sum_{i=1}^N g(\vec{x}^{(i)})}{\sum_{i=1}^N g(\vec{x}^{(i)})}$, in other words the weighted estimators are computed by using weights proportional to the objective function value of each individual in the selected set. In contrast with similar approaches (Hu et al., 2012; Yunpeng et al., 2006), they are some advantages of these derivations:

- A proportional weighting of the estimators avoid drastic changes when the individuals considerably differ in the objective function value. It is to say if a new individual with a large objective value is sampled, the exponential weights can concentrate the probability mass around this single individuals leading the algorithm to premature convergence.This less likeable when using proportional weights.
- A second advantage is that the minimum variance, which is bounded by a $\beta = \infty$ is not 0 for our approach, which is a significant advantage considering that naturally EDAs suffer of premature convergence and variance reduction (Hu et al., 2012; Yunpeng et al., 2006).

2.2 Annealing schedule 1

As can be seen in Equations 11, 12 and 13, the β value only affects the covariance matrix computation. The grade of impact of β over the covariance is highly related with $\sum_{i=1}^{N} g(\vec{x}^{(i)}$. It must be hold that $N/\beta < \sum_{i=1}^{N} g(\vec{x}^{(i)}$ in order to maintain a positive variance in the diagonal of the covariance matrix, on the other hand if $N/\beta << \sum_{i=1}^{N} g(\vec{x}^{(i)}$ its effect is diminished. Actually, when $\beta \to \infty$, $N/\beta \to 0$ and we get the minimum variance for our estimator. An interesting remark about this last setting is that the Normal with such minimum variance is not similar to a Dirac δ, while the corresponding Boltzmann actually is. Considering the arguments stated above Assume we propose a β value as shown in Equation 14.

$$\beta = (1-\gamma)N/\sum_{i=1}^{N} g(\vec{x}^{(i)}), \tag{14}$$

where $0 < \gamma < 1$ in order to fulfill the requirements discussed above. Hence, Equation 12 is rewritten as Equation 15.

$$\Sigma_* = \alpha \frac{\sum_{i=1}^{N} g(\vec{x}^{(i)})(\vec{x}^{(i)} - \vec{\mu})(\vec{x}^{(i)} - \vec{\mu})^t}{\sum_{i=1}^{N} g(\vec{x}^{(i)})} \tag{15}$$

For $\alpha = 1/\gamma$, recalling that $0 < \gamma < 1$, in consequence $1 < \alpha < \infty$. The first schedule proposed is to set α according historical improvements as follows:

For a maximization case.
if ($f_{best}^t > f_{best}^{t-1}$)
  $\alpha^t = 1.1\alpha^t$
else
  $\alpha^t = 0.9\alpha^t$
if ($\alpha^t > 2$)
  $\alpha^t = 2$
if ($\alpha^t < 1$)
  $\alpha^t = 1$

This schedule increases the covariance matrix values if an improvement in the objective function of the elite individual f_{best} is detected. Otherwise the covariance matrix at least gets its minimum values ($\beta = \infty$. On the other hand, we prevent to have scaling factors greater than 0 in order to control the exploration. The reader can notice that $\alpha = 1$, which is equivalent to $\beta = \infty$, indeed is a weighted estimator of the covariance matrix using weights proportional to the objective value, hence it captures the structure of the population favoring the most promising solutions.

2.3 Annealing schedule 2

In the first annealing schedule above, α in Equation 1 is increased or decreasing by multiplying it by a factor, hence it is actually increased/decreased in an exponential way with the number of improvements of the objective function. In this second proposal we use that $\alpha = 1/\gamma$, then gamma will be modified in a linear way with the improvements. Another difference between the two schedules is that the first schedule uses the improvements over the best objective value found so far, while this second schedule verifies the improvements over the selected set.Notice that updating β in Equation 13 is equivalent to update α in Equation 15 and equivalent to update γ considering that $\alpha = 1/\gamma$. According to the arguments in the Subbsection above, $0 < \gamma <= 1$, $\gamma = 1$ corresponds to the minimum variance ($\beta = \infty$). The updating of γ proceeds as follows:

- Let be M_s the number of selected individuals that are preserved from the current generation to the next one. Remember that the selection is performed over the union of the current population and the last selected set.
- Define a number of partitions of the interval $[0,1]$ as n_p. For our experiments $n_p = 31$.
- If $M_s/N > 0.5$ then $\gamma = \gamma - 1/n_p$, otherwise $\gamma = \gamma - 1/np$. Recall than N is the population size.
- If $\gamma < 1/np$ then $\gamma = 1/np$. If $\gamma > 1$ then $\gamma = 1$.

3 THE BOLTZMANN ESTIMATION OF MUTIVARIATE NORMAL ALGORITHM

The Boltzmann Estimation of Normal Multivariate Algorithm (BEMNA) is presented in Table 1. The lines marked with an asterisk will be explained in more detail below. The BEMNA starts with random population, the it is evaluated and half of the population with the best individuals are selected. The selected set objective function is used to compute the weights for the parameter computation, μ and Σ then are computed using the selected set variable values and the weights. Finally, using the computed parameters a new population is simulated. From the second generation in advance the selected set is computed by using the current population and the last selected set.

In Step 6, beta is update as mentioned in Subsections 2.2 and 2.3. We contrast the results of both schedules. In Step 7, the approximation is computed according to Equations 11 and 12. In Step 8, we sample $n_{pop} - 1$ new individuals, considering that at least

BEMNA

1. Give the parameter and stopping criterion: $n_{pop} \leftarrow$ Number of individuals to be sample.
2. Uniformly generate the initial population P_0, set $t = 0$.
3. While stopping criterion is not met
4. Evaluate the population.
5. Let be X^S the selected set, which are the best $n_{pop}/2$ individuals.
6. $t \leftarrow t+1$
7. *Update β^t.
8. *Compute the approximation to μ^t and Σ^t
9. *Sample the new population X^t using μ^t and Σ^t.
10. Select the best $n_{pop}/2$ individual from $X^t \cup X^S$ the union of X^t and the last selected set X^S.
11. Return the elite individual as the best approximation to the optimum.

Table 1: Pseudo-code of the BEMNA

the elite individual from the previous generation must survive.

A possible issue well known in Normal multivariate EDAs, is that it is possible that the covariance matrix present negative eigenvalues, due to numerical errors (Dong and Yao, 2008). In such a case we apply the following repairing scheme:

*L*et be X the matrix of eigenvectors of Σ by columns, and Λ a diagonal matrix with the corresponding eigenvalues, sorted from in decreasing order. And n the number of dimensions.

while $\Lambda_{n,n} < 0$
  $\lambda = \Lambda_{n,n}$
  For $i = 1..n$
    $\Lambda_{i,i} = \Lambda_{i,i} + \lambda$
  $\Sigma = X\Lambda X^t$
  Decompose Σ in X and Λ.

The repairing method is not called most of the time, and it is quite rare that it performs more than 1 iteration to fix the covariance matrix.

4 RESULTS

In this section we provide statistical results over well known test problems. We apply our proposal to the test in Table 2, in 30 dimensions using 15 independent runs for each function. The results are presented int the following subsections, divided by the annealing schedule used and the stopping criteria.

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

Table 2: Test problems. All of them are minimization problems. For applying the BEMNA they are converted to maximization and translated to positive as follows: $g(x) = -f(x) - min(-f(x)) + 1 \times 10^{-12}$. $f(x)$ is the function as it is shown in this table, and $g(x)$ is the one used in tha algorithm in Table 1.

4.1 Test problems with Schedule 1 and evaluations as stopping criterion

This subsection presents results using the first annealing schedule presented in Subsection 2.2. And using as stopping criterion when either: a maximum number of 3×10^5 evaluations is reach or the distance to the optimum is less than 1×10^{-6}. The results are presented in Table 3. When 0 is reported it means a value less than 1×-16

4.2 Discussion about the first set of experiments

In Table 3 presents the results of 15 independent executions for the set of problems in Table 2. Most of the problems, except the Rosenbrock function, are successfully solve by the BEMNA with a precision less than $1e-6$. Hence for most of them the important results is the number of function evaluations. In the case of the Rosenbrock it wort to notice that actually

Table 3: Objective function values from 15 independent executions. These results were obtained using the first annealing schedule and maximum 3e5 evaluations, the algorithm is also stopped if the optimum approximation is closer than 10^{-6}. The results are sorted in the same order than the functions in Table 2.

Best	Worts	Mean	Median	SD
8.973e-4	1.359e-1	2.718e-2	9.136e-3	3.946e-2
7.108e-7	9.965e-7	8.462e-7	8.332e-7	9.644e-8
7.286e-7	9.800e-7	9.134e-7	9.390e-7	6.573e-8
6.325e-7	9.970e-7	8.737e-7	8.693e-7	1.037e-7
7.237e-7	9.952e-7	8.988e-7	9.552e-7	9.573e-8
5.583e-7	9.959e-7	8.590e-7	9.138e-7	1.254e-7
3.969e-7	9.719e-7	8.284e-7	8.676e-7	1.617e-7
7.286e-7	9.874e-7	8.825e-7	8.994e-7	7.299e-8
2.842e-7	9.841e-7	7.518e-7	8.282e-7	2.047e-7
0	0	0	0	2.582e-7
0	0	0	0	0

Table 4: Number of evaluations from 15 independent executions. These results were obtained using the first annealing schedule and maximum 3e5 evaluations, the algorithm is also stopped if the optimum approximation is closer than 10^{-6}. The results are sorted in the same order than the functions in Table 2.

Best	Worts	Mean	Median	SD
3e5	3e5	3e5	3e5	0e0
9.56e4	1.23e5	1.11e5	1.12e5	7.63e3
1.40e5	1.81e5	1.58e5	1.58e5	1.00e4
7.77e4	9.92e4	8.59e4	8.53e4	6.01e3
7.63e4	1.05e5	9.01e4	9.20e4	7.92e3
1.08e5	1.24e5	1.18e5	1.20e5	5.29e3
1.28e5	1.65e5	1.48e5	1.48e5	1.07e4
1.20e5	1.44e5	1.33e5	1.33e5	7.24e3
3.64e4	5.21e4	4.27e4	4.31e4	4.07e3
1.06e5	1.30e5	1.17e5	1.14e5	6.74e3
1.81e5	2.20e5	2.00e5	2.01e5	1.03e4

the algorithm is not trapped in a local minimum, because the results are close to 0, with a value less than $1e-1$, that means that the algorithm is capable of displacing the search distribution to the optimum region, but it needs more computational effort to get it, as can be seen in the next set of experiments 4.3. In Table 4 we can see that all the problems, with exception of the Rosenbrock function, can be successfully solve with a similar computational cost. Even though they have different characteristics, for example the Ackley and Griewank functions are multimodal, but the algorithm does not increase significantly the number of function evaluation in contrast with the others. In the same way, all the convex problems as the ellipsoid, Two Axes, etc, with exception of the sphere, must adapt in a different way the covariance matrix, in Table 4 it can be notices that the BENMA does not present any inconvenience to adequately adapt the covariance matrix as demanded by the problem.

Table 5: Objective function values from 15 independent executions. These results were obtained using the first annealing schedule and maximum 6e5 evaluations, the algorithm is also stopped if the optimum approximation is closer than 10^{-6}. The results are sorted in the same order than the functions in Table 2.

Best	Worts	Mean	Median	SD
4.623e-7	9.959e-7	8.821e-7	9.530e-7	1.473e-7
5.269e-7	9.880e-7	8.563e-7	8.989e-7	1.405e-7
7.652e-7	9.938e-7	9.202e-7	9.659e-7	7.885e-8
6.471e-7	9.863e-7	8.913e-7	9.122e-7	9.044e-8
6.080e-7	9.979e-7	8.623e-7	8.910e-7	1.189e-7
7.754e-7	9.714e-7	8.716e-7	8.655e-7	6.812e-8
5.927e-7	9.934e-7	8.488e-7	8.439e-7	1.047e-7
6.653e-7	9.984e-7	8.519e-7	8.356e-7	1.226e-7
2.132e-7	9.740e-7	6.300e-7	6.590e-7	2.504e-7
0	0	0	0	0
0	0	0	0	0

Table 6: Number of evaluations from 15 independent executions. These results were obtained using the first annealing schedule and maximum 3e5 evaluations, the algorithm is also stopped if the optimum approximation is closer than 10^{-6}. The results are sorted in the same order than the functions in Table 2.

Best	Worts	Mean	Median	SD
3.19e5	3.60e5	3.41e5	3.38e5	1.34e4
9.02e4	1.22e5	1.09e5	1.09e5	8.03e3
1.32e5	1.73e5	1.57e5	1.58e5	1.08e4
7.50e4	1.07e5	8.84e4	8.58e4	9.57e3
7.50e4	1.11e5	9.01e4	9.11e4	8.96e3
1.12e5	1.32e5	1.22e5	1.22e5	6.19e3
1.38e5	1.58e5	1.50e5	1.50e5	5.66e3
1.23e5	1.41e5	1.33e5	1.33e5	5.00e3
3.41e4	5.61e4	4.34e4	4.36e4	5.59e3
1.03e5	1.38e5	1.15e5	1.14e5	8.85e3
1.78e5	2.17e5	1.91e5	1.92e5	1.07e4

4.3 Test problems with Schedule 1 and approximation to the optimum as stopping criterion

This subsection presents results using the first annealing schedule presented in Subsection 2.2. And using as stopping criterion a distance to the optimum less than 1×10^{-6}. The results are presented in Table 5. When 0 is reported it means a value less than 1×-16. The main difference with the section above is the Rosenbrock function, when using more than 3e5 evaluations it can achieve the desired precision. Notice in Table 6, that the number of evaluation is not considerably increased, with at most 6e4 evaluations more, the algorithm can reach a precision less than 1e-6.

Table 7: Objective function values from 15 independent executions. These results were obtained using the second annealing schedule and maximum 3e5 evaluations, the algorithm is also stopped if the optimum approximation is closer than 10^{-6}. The results are sorted in the same order than the functions in Table 2.

Best	Worts	Mean	Median	SD
8.43e-7	9.90e-7	9.54e-7	9.71e-7	4.44e-8
7.86e-7	9.97e-7	8.92e-7	9.10e-7	7.30e-8
6.89e-7	9.94e-7	9.13e-7	9.56e-7	9.62e-8
7.44e-7	9.97e-7	9.31e-7	9.51e-7	7.81e-8
8.25e-7	9.88e-7	9.01e-7	8.96e-7	5.15e-8
8.14e-7	9.99e-7	9.32e-7	9.40e-7	5.44e-8
3.42e-7	9.89e-7	8.40e-7	8.73e-7	1.80e-7
8.68e-7	9.89e-7	9.26e-7	9.29e-7	3.45e-8
7.05e-7	9.85e-7	9.17e-7	9.35e-7	7.49e-8
6.72e-7	9.95e-7	8.87e-7	9.13e-7	1.09e-7
6.36e-7	9.98e-7	8.72e-7	9.27e-7	1.28e-7

Table 8: Number of evaluations from 15 independent executions. These results were obtained using the second annealing schedule and maximum 3e5 evaluations, the algorithm is also stopped if the optimum approximation is closer than 10^{-6}. The results are sorted in the same order than the functions in Table 2.

Best	Worts	Mean	Median	SD
1.25e5	1.28e5	1.26e5	1.26e5	9.30e2
8.58e4	8.74e4	8.64e4	8.63e4	4.63e2
5.66e4	5.85e4	5.77e4	5.77e4	5.45e2
9.97e4	1.02e5	1.01e5	1.01e5	6.21e2
7.15e4	7.34e4	7.26e4	7.28e4	6.61e2
6.09e4	6.31e4	6.19e4	6.20e4	6.08e2
2.78e4	2.95e4	2.88e4	2.87e4	5.30e2
8.20e4	8.41e4	8.28e4	8.27e4	4.79e2
9.55e4	9.72e4	9.64e4	9.64e4	5.84e2
8.48e4	8.77e4	8.62e4	8.62e4	8.43e2
2.20e5	2.59e5	2.43e5	2.43e5	1.06e4

4.4 Discussion about the second set of experiments

In Table 5 we present the very same set of problems but using 6*e*5 as maximum number of evaluations. As can be seen all the problems solved in the first set of experiments are solved in this one, in addition the Rosenbrock function is successfully solved the 15 cases with the precision desired. It means that, even though we must intend to improve the efficiency of the BEMNA by reducing the number of function evaluations, the algorithm is quite effective, and can solve problems that similar approaches can not (Yunpeng et al., 2006), such as the Rosenbrock function.

4.5 Test problems with Schedule 2, using number of evaluations and optimum approximation as stopping criterion

This subsection presents results using the second annealing schedule presented in Subsection 2.3. We use as stopping criterion a distance to the optimum less than 1×10^{-6} or when the maximum number of evaluations 3e5 is reached. The results for the objective function value is shown in Table 7, and for the number of evaluations in Table 8.

4.6 Discussion about the third set of experiments

As mentioned in the last section, the first schedule can solve effectively the set of problems, but it takes more than 3e5 which is the number of function evaluations in (Yunpeng et al., 2006) for 10-dimensional problems. In this set of experimets, we intend to find solutions with precision less than $1e-6$, using less than 3e5 evaluations. As can be seen they are successfully solved, as it is shown in Tables 7 and 8. These results shows that the second annealing schedule proposed is more convenient for these problems. the extra computational effort of tracking the surviving individuals deliver an excellent payoff of a 63% reduction in the number of evaluations, in the Rosenbrok problem.

5 CONCLUSIONS

The main contribution of this proposal are the derivation of formulae for computing the parameters of an adequate Normal multivariate distribution for searching the optima, and the introduction of simple annelaing schedules for updating the β value.

The derived formulae for computing the search distribution use the objective function value as a linear factor for estimating weighted parameters. The linear weights avoid to prematurely collapse the probability mass around a single solution, preventing from premature convergence. In addition, this fashion of parameter estimation produces a softer change in the structure of the covariance matrix between consecutive generations, in contrast with the exponential weights used in similar approaches (Yunpeng et al., 2006). The advantage of using linear weights, even with a fixed β value, are well documented in (Valdez et al., 2013), where similar formulae are used for the univariate case. Our proposal combines the conviniences of the linear weights with simple annealing schedules to regulate the exploration of the algorithm.

The profit of using the linear weights and the annealing schedules is evidenced by statistical results presented in Section 4. These results show that our

proposal solves the Rosenbrock problem which is not solved by similar algorithms, as well as the other problems with an inferior computational cost than the used in **??** and (Valdez et al., 2013). We recommend to the reader to contrast the results with the cited articles in order to verify the conveniences of our approach. The explanation of this superior performance is that the annealing schedules proposed prevent from a drastic variance reduction, besides, if a suboptimal individual or inadequate β parameter is used in some generation, the impact over the whole optimization process is less drastic when using linear weights than in those proposals that use exponential ones (Yunpeng et al., 2006). In other words, our proposal is more robust to fake optimum and inadequate annealing schedules than similar ones.

Future work contemplate to propose additional enhancement techniques to be applied over the current BEMNA for reducing the population size as well as the number of function evaluations. Moreover, we will explore to unifiy a Boltzmann based EDA framework, which can be applied for continuos and discrete optimization, using multivariate or univariate search distribution models.

REFERENCES

Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience.

Dong, W. and Yao, X. (2008). Unified Eigen analysis on multivariate Gaussian based estimation of distribution algorithms. *Information Sciences*, 178(15):215–247.

Grahl, J., Bosman, P. A. N., and Minner, S. (2007). Convergence phases, variance trajectories, and runtime analysis of continuos EDAs. In *GECCO '07: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 516–522. ACM.

Hu, J., Wang, Y., Zhou, E., Fu, M. C., and Marcus, S. I. (2012). A survey of some model-based methods for global optimization. In *Optimization, Control, and Applications of Stochastic Systems*, pages 157–179. Springer.

Larrañaga, P., Etxeberria, R., Lozano, J. A., and Peña, J. M. (2000). Optimization in continuous domains by learning and simulation of gaussian networks. Technical Report EHU-KZAA-IK-4/99, University of the Basque Country.

Larraaga, P. (2002). A review on estimation of distribution algorithms. In Larraaga, P. and Lozano, J., editors, *Estimation of Distribution Algorithms*, volume 2 of *Genetic Algorithms and Evolutionary Computation*, pages 57–100. Springer US.

Mahnig, T. and Muhlenbein, H. (2001). A new adaptive boltzmann selection schedule sds. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pages 183–190. IEEE.

Müehlenbein[1], H., Bendisch[1], J., and Voight[2], H.-M. (1996). From recombination of genes to the estimation of distributions ii. continuous parameters.

Muhlenbein, H. and Mahnig, T. (1999). The factorized distribution algorithm for additively decomposed functions. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1. IEEE.

Mühlenbein, H., Mahnig, T., and Rodriguez, A. O. (1999). Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):215–247.

Mühlenbein, H., Mahnig, T., and Rodriguez, A. O. (1999). Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):215–247.

Mhlenbein, H. (2012). Convergence theorems of estimation of distribution algorithms. In Shakya, S. and Santana, R., editors, *Markov Networks in Evolutionary Computation*, volume 14 of *Adaptation, Learning, and Optimization*, pages 91–108. Springer Berlin Heidelberg.

Ochoa, A. (2010). Opportunities for expensive optimization with estimation of distribution algorithms. In *Computational Intelligence in Expensive Optimization Problems*, volume 2, pages 193–218. Springer.

Shapiro, J. L. (2006). Diversity loss in general estimation of distribution algorithms. In *Parallel Problem Solving from Nature-PPSN IX*, pages 92–101. Springer.

Valdez, S. I., Hernández, A., and Botello, S. (2013). A boltzmann based estimation of distribution algorithm. *Information Sciences*, 236:126–137.

Yunpeng, C., Xiaomin, S., and Peifa, J. (2006). Probabilistic modeling for continuous eda with boltzmann selection and kullback-leibeler divergence. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 389–396. ACM.