A Boltzmann Multivariate Estimation of Distribution Algorithm for Continuous Optimization

Ignacio Segovia-Domínguez, S. Ivvan Valdez and Arturo Hernández-Aguirre
Department of Computer Science, Center for Research in Mathematics, Guanajuato, Mexico

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

Abstract: This paper introduces an approach for continuous optimization using an Estimation of Distribution Algorithm (EDA), based on the Boltzmann 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 Boltzmann distribution directly for sampling is not possible because it requires the computation of the objective function values in the complete search space. This work presents an approximation to the Boltzmann function by a multivariate Normal distribution. Formulae for computing the mean and covariance matrix are derived by minimizing the Kullback-Leibler divergence. The proposed EDA is competitive and often superior to similar algorithms as it is shown by statistical results reported here.

1 INTRODUCTION

The 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. In fact, 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 (Larrañaga, 2002; Larrañaga et al., 2000; Dong and Yao, 2008; Segovia-Dominguez et al., 2013). The parameters of the Normal density are estimated by using maximum likelihood estimators (MLEs) over the selected set, which is determined by the truncation method; usually 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 in fitness values.
- It is a well-known issue that the variance in estimation of distribution algorithms is often less than required, hence, the MLE variance estimator is not the most adequate for searching the optimum (Shapiro, 2006; Grahl et al., 2007).

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

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

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

Segovia-Domínguez I., Valdez S. and Hernández-Aguirre A..
A Boltzmann Multivariate Estimation of Distribution Algorithm for Continuous Optimization.
DOI: 10.5220/0005079902510258
In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications* (ECTA-2014), pages 251-258
ISBN: 978-989-758-052-9

As can be noticed in Equation (1) the objective function is used as an energy function directly. In practical approaches, the Boltzmann distribution cannot be used directly for sampling because it is necessary to know the objective function in the whole domain. That is the reason why the parameters of a density function are computed by minimizing a measure between the parametric distribution and the Boltzmann distribution, for instance, the Kullback-Leibler divergence (Ochoa, 2010; Yunpeng et al., 2006; Valdez et al., 2013).

There are remarkable challenges to consider for the designing of EDAs based on the Boltzmann distribution:

- To choose an adequate β parameter in Equation (1). Usually β depends on the time or is dynamic during the optimization process. The process 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 parameter estimators for the approximated distribution. Some approaches (Hu et al., 2012; Yunpeng et al., 2006) have derived formulae for estimating parameters of a distribution which approximate the Boltzmann density, 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 the second option is to obtain formulae which do not impact the estimators as drastically.
- The last two issues are also related with the mentioned variance reduction which is a common issue in EDAs (Shapiro, 2006).

Our proposal intends to tackle the challenges just mentioned. Accordingly, this paper presents 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 are robust or at least are not impacted as drastically as the exponential function used in other approaches.
- The novel annealing schedules tackle the variance reduction problem; hence, these are mechanisms to avoid the premature convergence of the algorithm.

The organization of the paper is as follows: Section 2 presents the derivation of the formulae 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 is devoted to testing the proposed EDA on well-known test functions. Also, a comparison to another algorithm from literature is performed. Finally, Section 5 provides some concluding remarks.

2 APPROXIMATING THE BOLTZMANN DISTRIBUTION BY THE NORMAL MULTIVARIATE DISTRIBUTION

This section introduces the formulae to estimate the mean vector and covariance matrix, 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 $\vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(N)}$ 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).

$$Q_x = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left\{ -\frac{1}{2}(\vec{x}-\vec{\mu})^t \Sigma^{-1} (\vec{x}-\vec{\mu}) \right\} \quad (2)$$

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

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). Here, the Kullback-Leibler Divergence presented in Equation (4), $K_{QP} = D_{KL}(Q_x||P_x)$, is used as statistical measure between probability distributions.

$$K_{QP} = \int Q_x \log \frac{Q_x}{P_x} d\vec{x} \quad (4)$$

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

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

Notice that K_{QP} can be rewrite as

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

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, the partial derivatives are calculated; 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 vector and covariance matrix are obtained by making the derivatives equal to 0, as in Equations (9) and (10), and solving for $\vec{\mu}$ and Σ respectively.

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

The following equivalences were used to obtain Equations (9) and (10):

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

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

Finally, for estimating the parameters using the observations $\vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(N)}$ of the random variable $\vec{X}$, a numerical stochastic approximation by the Monte Carlo method is computed, as shown in Equations (11) and (12). These two equations are the estimators that approximate 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 use weights defined by $\frac{g(\vec{x}^{(i)})}{\sum_{i=1}^N g(\vec{x}^{(i)})}$. 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), there are some advantages of these derivations:

- A proportional weight of the estimators avoids drastic changes when the individuals considerably differ in the objective function value. It is to say that if a new individual with a large objective value is sampled, the exponential weights can concentrate the probability mass around this single individual, leading the algorithm to premature convergence. This is less desirable 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 from premature convergence and variance reduction (Hu et al., 2012; Yunpeng et al., 2006).

The energy function $g(\vec{x})$ must be positive or equal to zero in the domain. However, the objective function $\mathcal{F}(\vec{x})$ might be negative, considering that this is a maximization/minimization problem. In order to construct a valid energy function for the continuous minimization problem, throughout this paper the $g(\vec{x}_i)$ value is computed as $g(\vec{x}_i) = -\mathcal{F}(\vec{x}_i) - \min(-\mathcal{F}(\vec{x}_i)) + 10^{-12}$.

2.2 Annealing Schedule 1

As 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 $\sum_{i=1}^{N} g(\vec{x}^{(i)})$. Accordingly, $N/\beta < \sum_{i=1}^{N} g(\vec{x}^{(i)})$ must remain in order to maintain a positive variance in the diagonal of the covariance matrix, on the other hand if $N/\beta << \sum_{i=1}^{N} g(\vec{x}^{(i)})$ then its effect is diminished. As a consequence, the estimator reaches the minimum variance when $\beta \to \infty$ because $N/\beta \to 0$. An interesting remark about this last setting is that the Normal distribution with such a minimum variance is not similar to a Dirac δ, while the corresponding Boltzmann distribution actually is. Considering the arguments stated above, assume a β value as shown in Equation (14).

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

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 the minimization case:
if $(\mathcal{F}_{best}^{(t)} < \mathcal{F}_{best}^{(t-1)})$ $\alpha = 1.1\alpha$
else $\alpha = 0.9\alpha$
if $(\alpha > 2)$ $\alpha = 2$
if $(\alpha < 1)$ $\alpha = 1$

This schedule increases the covariance matrix values if an improvement in the objective function of the elite individual $\mathcal{F}_{best}$ is detected. Otherwise, the covariance matrix at least gets its minimum values ($\beta = \infty$). On the other hand, we prevent having 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 it is a weighted estimator of the covariance matrix using weights proportional to the objective value. Hence, this captures the structure of the population favoring the most promising solutions.

2.3 Annealing Schedule 2

The second annealing schedule uses $\alpha = 1/\gamma$, where γ will be modified in a linear way with the improvements. This shows a difference versus the previous proposal because the first annealing schedule increases/decreases the α value in an exponential way. 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 updating α in Equation (15) and equivalent to updating γ considering that $\alpha = 1/\gamma$. According to the arguments in the Subsection above, $0 < \gamma \leq 1$, $\gamma = 1$ corresponds to the minimum variance ($\beta = \infty$). The updating of γ proceeds as follows:

- Let M_s be the number of selected individuals that are preserved from the current generation to the next one, throughout this paper known as the *survivor individuals*. Please note that the maximum number of survivor individuals is the sample size S_c.
- Define a number of partitions of the interval $[0,1]$ as n_p. For our experiments $n_p = 30$.
- If $M_s/S_c > 0.5$ then $\gamma = \gamma - 1/n_p$, otherwise $\gamma = \gamma + 1/n_p$.
- If $\gamma < 1/n_p$ then $\gamma = 1/n_p$. 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 Figure 1. The BEMNA starts with a random population, then 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, $\vec{\mu}$ and Σ then are computed using the selected set variable values and the weights. Finally, using the computed parameters a set of individuals is simulated. From the second generation in advance the new selected set is computed by using the current selected set and the simulated one.

```
BEMNA
1. $t \leftarrow 0$. Initialize $N_{pop}$, $N_{sel}$, $S_c$ and $\alpha/\gamma$ according to the annealing schedule.
2. $P^{(0)} \leftarrow$ Uniformly distributed population, where $P^{(0)} = \{\vec{x}_1, ..., \vec{x}_{N_{sel}}\}$.
3. Evaluate $\mathcal{F}(\vec{x}_i)$ and $g(\vec{x}_i)$.
4. Compute the estimates $\vec{\mu}_*$ and $\Sigma_*$ of $P^{(t)}$, by equations (11) and (15).
5. $P_S^{(t)} \leftarrow$ Sampling $S_c$ individuals from $Q(x;\vec{\mu}_*, \Sigma_*)$.
6. Evaluate $\mathcal{F}(\vec{x}_j)$ and $g(\vec{x}_j)$ of $P_S^{(t)}$.
7. Update $\alpha/\gamma$ according to the annealing schedule.
8. $P^{(t+1)} \leftarrow$ The best $N_{sel}$ individuals from $P^{(t)} \cup P_S^{(t)}$.
9. $t \leftarrow t+1$.
10. If stopping criterion is not reached return to step 4.
11. Return the elite individual in $P^{(t)}$ as the best approximation to the optimum.
```

Figure 1: Pseudo-code of the BEMNA.

3.1 Detailed Steps for the First Annealing Schedule

The recommended population size for this schedule is of $N_{pop} = 15d$ where d is the number of dimensions. Initial $\alpha = 1$. The number of selected individuals in Steps 2 and 8 is $N_{sel} = N_{pop}/2$. The parameter estimation for the multivariate Normal in Step 4 is done as explained in Section 2. In Step 5 the new sample is actually $P_S^{(t)}$ (which is different in the second schedule) of size N_{pop}. To update the α value the annealing schedule is performed as explained in Subsection 2.2. Please remember that for each α there is a corresponding β and vice versa.

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

Let L be the matrix of eigenvectors of Σ by columns, and Λ a diagonal matrix with the corresponding eigenvalues, in decreasing order, and d the number of dimensions.

```
while $\Lambda_{d,d} < 0$
    $\lambda = -\Lambda_{d,d}$
    For $i = 1..d$
    $\Lambda_{i,i} = \Lambda_{i,i} + \lambda$
    $\Sigma = L\Lambda L^t$
    Decompose $\Sigma$ in $L$ and $\Lambda$
```

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

3.2 Detailed Steps for the Second Annealing Schedule

For the second annealing schedule the population size is $N_{pop} = \lceil (d+3)(1+d^{0.7}) \rceil$, while the sample size is $S_c = \lceil 2(1+d^{0.7}) \rceil$; taken from an empirical test. For this schedule the γ parameter is updated. Remember that there is a direct relation among β, α and γ; one of them can define the others. The initial gamma value is $\gamma^0 = 0.5 - 1/n_p$. In this schedule the number of selected individuals is the same as the population size $N_{sel} = N_{pop}$. In the first generation, the complete population becomes the selected set, as we do not have more individuals. In Step 5 a new sample $P_S^{(t)}$ of size S_c is simulated. In Step 7 the annealing schedule is performed as explained in Subsection 2.3 for updating γ. Finally in Step 8, a new selected set $P^{(t+1)}$ of size N_{sel} is obtained.

Table 1: Test problems (Larrañaga, 2002; Valdez et al., 2013). All of them are minimization problems. For applying the BEMNA these are converted to maximization and translated to positive as follows: $g(\vec{x}) = -\mathcal{F}(\vec{x}) - min(-\mathcal{F}(\vec{x})) + 1 \times 10^{-12}$. Where $\mathcal{F}(\vec{x})$ is the objective function and $g(\vec{x})$ is the energy function used in Figure 1. The minimum fitness value of all problems is 0 except for $\mathcal{F}_{12}$ where $\mathcal{F}_{12}^* = -d(d+4)(d-1)/6$.

$\mathcal{F}$	Name	Domain
$\mathcal{F}_1$	Sphere	$[-10, 5]^d$
$\mathcal{F}_2$	Tablet	$[-10, 5]^d$
$\mathcal{F}_3$	Ellipsoid	$[-10, 5]^d$
$\mathcal{F}_4$	Cigar	$[-10, 5]^d$
$\mathcal{F}_5$	Cigar Tablet	$[-10, 5]^d$
$\mathcal{F}_6$	Different Powers	$[-10, 5]^d$
$\mathcal{F}_7$	Parabolic Ridge	$[-10, 5]^d$
$\mathcal{F}_8$	Sharp Ridge	$[-10, 5]^d$
$\mathcal{F}_9$	Griewank	$[-600, 600]^d$
$\mathcal{F}_{10}$	Ackley	$[-32.768, 16.384]^d$
$\mathcal{F}_{11}$	Rosenbrock	$[-10, 5]^d$
$\mathcal{F}_{12}$	Trid	$[-d^2, d^2]^d$
$\mathcal{F}_{13}$	Brown	$[-1, 4]^d$
$\mathcal{F}_{14}$	Levy Montalvo 1	$[-20, 10]^d$
$\mathcal{F}_{15}$	Levy Montalvo 2	$[-20, 10]^d$
$\mathcal{F}_{16}$	Levy 8	$[-20, 10]^d$
$\mathcal{F}_{17}$	Pinter	$[-20, 10]^d$

4 EXPERIMENTS

This section provides statistical results from well-known test problems, see Table 1. The BEMNA, described in Figure 1, is tested in three different ways: 1) using the annealing schedule 1, 2) using the annealing schedule 2, and 3) comparing our proposal against a previous version with uncorrelated variables, taken from literature. The following subsections discuss each of these experiments.

Table 2: Statistical results in 30 dimensions of 11 test problems, from 15 independent executions. First row: objective values reached, and second row: number of function evaluations. If the obtained fitness value reaches the desired precision, i.e. $\mathcal{F} - \mathcal{F}^* < 1 \times 10^{-6}$, then this cell is boldfaced. (a) Schedule 1. (b) Schedule 2.

(a)

$\mathcal{F}$	**Best**	**Worst**	**Mean**	**Median**	**SD**
$\mathcal{F}_1$	**6.32e-7**	**9.97e-7**	**8.73e-7**	**8.69e-7**	1.03e-7
	7.77e4	9.92e4	8.59e4	8.53e4	6.01e3
$\mathcal{F}_2$	**7.23e-7**	**9.95e-7**	**8.98e-7**	**9.55e-7**	9.57e-8
	7.63e4	1.05e5	9.01e4	9.20e4	7.92e3
$\mathcal{F}_3$	**5.58e-7**	**9.95e-7**	**8.59e-7**	**9.13e-7**	1.25e-7
	1.08e5	1.24e5	1.18e5	1.20e5	5.29e3
$\mathcal{F}_4$	**3.96e-7**	**9.71e-7**	**8.28e-7**	**8.67e-7**	1.61e-7
	1.28e5	1.65e5	1.48e5	1.48e5	1.07e4
$\mathcal{F}_5$	**7.28e-7**	**9.87e-7**	**8.82e-7**	**8.99e-7**	7.29e-8
	1.20e5	1.44e5	1.33e5	1.33e5	7.24e3
$\mathcal{F}_6$	**2.84e-7**	**9.84e-7**	**7.51e-7**	**8.28e-7**	2.04e-7
	3.64e4	5.21e4	4.27e4	4.31e4	4.07e3
$\mathcal{F}_7$	**0.00**	**0.00**	**0.00**	**0.00**	2.58e-7
	1.06e5	1.30e5	1.17e5	1.14e5	6.74e3
$\mathcal{F}_8$	**0.00**	**0.00**	**0.00**	**0.00**	0.00
	1.81e5	2.20e5	2.00e5	2.01e5	1.03e4
$\mathcal{F}_9$	**7.10e-7**	**9.96e-7**	**8.46e-7**	**8.33e-7**	9.64e-8
	9.56e4	1.23e5	1.11e5	1.12e5	7.63e3
$\mathcal{F}_{10}$	**7.28e-7**	**9.80e-7**	**9.13e-7**	**9.39e-7**	6.57e-8
	1.40e5	1.81e5	1.58e5	1.58e5	1.00e4
$\mathcal{F}_{11}$	8.97e-4	1.35e-1	2.71e-2	9.13e-3	3.94e-2
	3.00e5	3.00e5	3.00e5	3.00e5	0.00
$\overline{\mathcal{F}}_{11}$	**4.62e-7**	**9.95e-7**	**8.82e-7**	**9.53e-7**	1.47e-7
	3.19e5	3.60e5	3.41e5	3.38e5	1.34e4

(b)

$\mathcal{F}$	**Best**	**Worst**	**Mean**	**Median**	**SD**
$\mathcal{F}_1$	**7.44e-7**	**9.97e-7**	**9.31e-7**	**9.51e-7**	7.81e-8
	9.97e4	1.02e5	1.01e5	1.01e5	6.21e2
$\mathcal{F}_2$	**8.25e-7**	**9.88e-7**	**9.01e-7**	**8.96e-7**	5.15e-8
	7.15e4	7.34e4	7.26e4	7.28e4	6.61e2
$\mathcal{F}_3$	**8.14e-7**	**9.99e-7**	**9.32e-7**	**9.40e-7**	5.44e-8
	6.09e4	6.31e4	6.19e4	6.20e4	6.08e2
$\mathcal{F}_4$	**3.42e-7**	**9.89e-7**	**8.40e-7**	**8.73e-7**	1.80e-7
	2.78e4	2.95e4	2.88e4	2.87e4	5.30e2
$\mathcal{F}_5$	**8.68e-7**	**9.89e-7**	**9.26e-7**	**9.29e-7**	3.45e-8
	8.20e4	8.41e4	8.28e4	8.27e4	4.79e2
$\mathcal{F}_6$	**7.05e-7**	**9.85e-7**	**9.17e-7**	**9.35e-7**	7.49e-8
	9.55e4	9.72e4	9.64e4	9.64e4	5.84e2
$\mathcal{F}_7$	**6.72e-7**	**9.95e-7**	**8.87e-7**	**9.13e-7**	1.09e-7
	8.48e4	8.77e4	8.62e4	8.62e4	8.43e2
$\mathcal{F}_8$	**6.36e-7**	**9.98e-7**	**8.72e-7**	**9.27e-7**	1.28e-7
	2.20e5	2.59e5	2.43e5	2.43e5	1.06e4
$\mathcal{F}_9$	**7.86e-7**	**9.97e-7**	**8.92e-7**	**9.10e-7**	7.30e-8
	8.58e4	8.74e4	8.64e4	8.63e4	4.63e2
$\mathcal{F}_{10}$	**6.89e-7**	**9.94e-7**	**9.13e-7**	**9.56e-7**	9.62e-8
	5.66e4	5.85e4	5.77e4	5.77e4	5.45e2
$\mathcal{F}_{11}$	**8.43e-7**	**9.90e-7**	**9.54e-7**	**9.71e-7**	4.44e-8
	1.25e5	1.28e5	1.26e5	1.26e5	9.30e2

4.1 Testing the BEMNA with Both Annealing Schedules

This subsection presents the results of the BEMNA in 30 dimensions. This set of problems is taken from the literature: 8 unimodal, 2 multimodal and the generalized Rosenbrock functions. The population size, number of samples and similar issues are discussed in subsections 2.2 and 2.3. In addition, the algorithm is stopped when either: a maximum number of 3×10^5 evaluations is reached or the precision to the optimum value is $\mathcal{F} - \mathcal{F}^* < 1 \times 10^{-6}$, except for problem $\overline{\mathcal{F}}_{11}$ where the maximum number is 6×10^5 evaluations.

The results are presented in Table 2. The value 0 is reported if $\mathcal{F} - \mathcal{F}^* < 1 \times 10^{-16}$. For each problem the statistics are presented in two rows. The first one summarizes the fitness values reached whilst the second one shows the needed number of evaluations. If the obtained fitness value reaches the desired precision, i.e. $\mathcal{F} - \mathcal{F}^* < 1 \times 10^{-6}$, then this cell is boldfaced. Here, $\mathcal{F}^*$ is the optimum value.

4.1.1 Results Using the Annealing Schedule 1

The Table 2-(a) presents the results of 15 independent executions for the set of 10 unimodal functions and the Rosenbrock problem ($\mathcal{F}_{11}$). Most of the problems, except the Rosenbrock function, are successfully solved by the BEMNA with a precision less than $1e-6$. Actually, the column of worst values shows a perfect success rate, except for the Rosenbrock problem. Therefore for most of the problems the important result is the number of function evaluations.

Also, note that all the problems, with exception of the Rosenbrock function, can be successfully solved with a similar computational cost. Even though they have different characteristics, for example the Ackley and Griewank functions are multimodal, the algorithm does not significantly increase the number of function evaluations in contrast with the other problems. Also, the covariance matrix is adapted according to the function; as seen in all the convex problems.

In the case of $\mathcal{F}_{11}$ it is worth noting that the algorithm is not trapped in a local minimum, because the reached fitness values are less than 0.1. This means that the algorithm is capable of displacing the search distribution to the optimum region. An extra experiment in this problem, please see $\overline{\mathcal{F}}_{11}$, shows that the BEMNA (using the first annealing schedule) needs more computational effort to solve the Rosenbrock problem.

4.1.2 Results Using the Annealing Schedule 2

Similar to the last subsection, Table 2-(b) shows the results of our proposal, but using the second annealing schedule. The extra computational effort of tracking the surviving individuals delivers an excellent payoff of a 63% reduction in the number of evaluations, for the Rosenbrock problem; whilst the rest of the test problems may or may not be improved. An inspection in the mean values of the function evaluations show some differences in comparison with the BEMNA using the first annealing schedule. In fact, our proposal using the second annealing schedule needed less computational effort to solve 7 out of 10 test problems; $\mathcal{F}_2$-$\mathcal{F}_5$, $\mathcal{F}_7$, $\mathcal{F}_9$ and $\mathcal{F}_{10}$.

These results show that this schedule is more convenient than the previous one. Even though the number of function evaluations are reduced, the BEMNA is quite effective and can solve problems that similar approaches can not (Yunpeng et al., 2006), such as the Rosenbrock function. Despite the differences between both schedules, we can conclude that the BENMA does not present any inconvenience to adequately adapt the covariance matrix as demanded by the problem.

4.2 Comparison Versus the BUMDA

Since the Boltzmann distribution has been widely used in evolutionary computation, a comparison against a related method from literature is desirable. In order to make a comparison versus the state of the art in evolutionary computing, a successful algorithm in this branch is selected: Boltzmann Univariate Marginal Distribution Algorithm (BUMDA), see (Valdez et al., 2013). Our proposal is capable of modeling dependency between variables. According to this property, a suitable set of problems is chosen: $\mathcal{F}_{11}$-$\mathcal{F}_{17}$. Here, we use the BEMNA with the second annealing schedule.

Table 3 contrasts the error $\mathcal{F} - \mathcal{F}^*$ reached for each algorithm. For each problem there are three measures: 1) the first row is the percentage of success rate, 2) the second row is the mean and standard deviation of reached fitness values, 3) the third row is the mean and standard deviation of needed evaluations of function. In addition, the performance between both algorithms is verified by a non-parametric bootstrap test. Here, the hypothesis is based in the mean value μ. So, if the null hypothesis $H_0 : \mu_1 = \mu_2$ is rejected there is statistical evidence to accept differences between both algorithms; this case is marked in bold.

Table 3: Percentage of success rate, reached fitness values and needed number of evaluations (mean and standard deviation) for each algorithm. The last column shows a non-parametric bootstrap test. If ρ is boldface there are statistical evidence of difference between both algorithms. (a) 10 dimensions. (b) 30 dimensions.

(a)

$\mathcal{F}$	BEMNA	BUMDA	ρ
$\mathcal{F}_{11}$	**100.00**	0.00	
	7.29e-7±2.10e-7	8.08e+0±7.13e-2	**1.00e-4**
	1.59e+4±1.20e+3	3.00e+5±0.00e+0	**1.00e-4**
$\mathcal{F}_{12}$	**100.00**	0.00	
	8.17e-7±1.29e-7	6.58e+1±1.56e+1	**1.00e-4**
	8.04e+3±1.72e+2	3.00e+5±0.00e+0	**1.00e-4**
$\mathcal{F}_{13}$	**100.00**	**100.00**	
	8.17e-7±1.52e-7	7.03e-7±1.92e-7	7.63e-2
	6.04e+3±1.89e+2	1.17e+4±2.99e+2	**1.00e-4**
$\mathcal{F}_{14}$	**100.00**	**100.00**	
	8.54e-7±1.04e-7	7.43e-7±1.87e-7	5.26e-2
	5.36e+3±1.68e+2	1.12e+4±2.49e+2	**1.00e-4**
$\mathcal{F}_{15}$	**100.00**	**100.00**	
	7.27e-7±1.85e-7	7.14e-7±1.94e-7	8.45e-1
	5.90e+3±2.48e+2	1.28e+4±3.61e+2	**1.00e-4**
$\mathcal{F}_{16}$	**100.00**	**100.00**	
	7.80e-7±1.81e-7	6.57e-7±2.07e-7	8.34e-2
	5.71e+3±1.47e+2	1.15e+4±2.22e+2	**1.00e-4**
$\mathcal{F}_{17}$	80.00	**100.00**	
	7.31e+0±1.52e+1	7.95e-7±1.90e-7	7.39e-2
	2.71e+4±3.78e+4	1.91e+4±6.98e+2	3.73e-1

(b)

$\mathcal{F}$	BEMNA	BUMDA	ρ
$\mathcal{F}_{11}$	**100.00**	0.00	
	9.27e-7±7.11e-8	2.80e+1±7.56e-2	**1.00e-4**
	2.52e+5±9.92e+3	3.00e+5±0.00e+0	**1.00e-4**
$\mathcal{F}_{12}$	**100.00**	0.00	
	9.12e-7±7.13e-8	5.00e+3±1.08e+3	**1.00e-4**
	8.40e+4±7.91e+2	3.00e+5±0.00e+0	**1.00e-4**
$\mathcal{F}_{13}$	**100.00**	93.33	
	8.86e-7±9.44e-8	1.08e-6±8.72e-7	4.67e-1
	5.62e+4±1.18e+3	4.25e+4±7.13e+4	4.42e-1
$\mathcal{F}_{14}$	**100.00**	**100.00**	
	8.85e-7±8.55e-8	8.47e-7±1.29e-7	3.35e-1
	4.51e+4±6.21e+2	**2.05e+4±2.20e+2**	**1.00e-4**
$\mathcal{F}_{15}$	**100.00**	**100.00**	
	9.23e-7±6.00e-8	**7.99e-7±9.51e-8**	**4.00e-4**
	5.47e+4±6.58e+2	**2.33e+4±3.29e+2**	**1.00e-4**
$\mathcal{F}_{16}$	**100.00**	**100.00**	
	9.46e-7±4.87e-8	**8.60e-7±1.04e-7**	**7.60e-3**
	5.11e+4±6.23e+2	**2.21e+4±3.29e+2**	**1.00e-4**
$\mathcal{F}_{17}$	**40.00**	0.00	
	4.48e+1±5.38e+1	**1.57e-2±1.66e-2**	**5.40e-3**
	2.14e+5±1.10e+5	3.00e+5±0.00e+0	**6.20e-3**

The BEMNA shows a better performance than the BUMDA in most of the test problems in 10 dimensions. Also, notice that the BEMNA solves all the problems using fewer evaluations than the BUMDA. On the other hand, for 30 dimensional problems the BUMDA displays a better performance than our proposal in $\mathcal{F}_{14} - \mathcal{F}_{16}$. It is an expected result because these problems have weakly correlated variables. However, our approach effectively solved the hardest problems in 30 dimensions: $\mathcal{F}_{11}, \mathcal{F}_{12}$ and $\mathcal{F}_{17}$.

5 CONCLUSIONS

The main contributions of this proposal are the derivation of formulae for computing the parameters of an adequate Multivariate Normal Distribution for locating the optimum, and the introduction of simple annealing 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 prematurely collapsing probability mass around a single solution, preventing 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 to the exponential weights used in similar approaches (Yunpeng et al., 2006). The advantage of using linear weights, even with a fixed β value, is 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 advantage of using the linear weights and the annealing schedules is evidenced by statistical results presented in Section 4. Furthermore, the results demonstrate that the BEMNA effectively solves the Rosenbrock problem, which is not solved by similar algorithms, (Yunpeng et al., 2006) and (Valdez et al., 2013); as well as the other problems using an inferior computational cost.

Future work intends 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 new ways to use this approach in evolutionary computation.

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

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.

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.

Mühlenbein, 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.

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.

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.

Segovia-Dominguez, I., Hernandez-Aguirre, A., and Diharce, E. V. (2013). The gaussian polytree eda with copula functions and mutations. In *EVOLVE*, volume 447 of *Studies in Computational Intelligence*, pages 123–153. Springer Berlin Heidelberg.

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.