A Boltzmann based estimation of distribution algorithm

ARTICLE *in* INFORMATION SCIENCES · JULY 2013

Impact Factor: 4.04 · DOI: 10.1016/j.ins.2013.02.040

3 AUTHORS, INCLUDING:

Sergio Ivvan Valdez Peña
Centro de Investigación en Matemáticas (CIM…
29 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Arturo Hernandez-Aguirre
Centro de Investigación en Matemáticas (CIM…
138 PUBLICATIONS **883** CITATIONS
SEE PROFILE

A Boltzmann based Estimation of Distribution Algorithm

S. Ivvan Valdez[a,*], Arturo Hernández[a], Salvador Botello[a]

[a] *Centre for Research in Mathematics (CIMAT) A.C., Jalisco S/N, Guanajuato, Gto., México. C.P. 36240*

Abstract

This paper introduces a new approach for estimation of distribution algorithms called the Boltzmann Univariate Marginal Distribution Algorithm (BUMDA). It uses a Normal-Gaussian model to approximate the Boltzmann distribution, hence, formulae for computing the mean and variance parameters of the Gaussian model are derived from the analytical minimization of the Kullback-Leibler divergence. The resulting formulae explicitly introduces information about the fitness landscape for the Gaussian parameters computation, in consequence, the Gaussian distribution obtains a better bias to sample intensively the most promising regions than simply using the maximum likelihood estimator of the selected set. In addition, the BUMDA formulae needs only one user parameter. Accordingly to the experimental results, the BUMDA excels in its niche of application. We provide theoretical, graphical and statistical analysis to show the BUMDA performance contrasted with state of the art EDAs.

Keywords: EDA, Boltzmann distribution, Kullback-Leibler, Normal distribution, Selection Methods

Plesea refer this article as: doi:10.1016/j.ins.2013.02.040

*Tel. +52 (473) 7327155, Fax +52 (473) 7325749

Email addresses: ivvan@cimat.mx (S. Ivvan Valdez), artha@cimat.mx (Arturo Hernández), botello@cimat.mx (Salvador Botello)

1. Introduction

The Estimation of Distribution Algorithms (EDAs) were derived from probabilistic modeling of Genetic Algorithms (GAs) [12] [1]. Nevertheless, EDAs and other evolutionary algorithms differ in the strategy taken for approximating the optimum. One of the main differences between EDAs and evolutionary algorithms such as GAs, is the way the population is regenerated: while GAs use a subset of the fittest individuals in the current population (selected set), EDAs use a probability distribution, called the search distribution, to sample new candidate solutions. Hence, the search strategy in EDAs is to increment the probability of sampling the optimum from the search distribution. Practical EDAs assume that the most promising regions for finding the optimum are such regions which contain the individuals with the greatest objective values (for a maximization case). Therefore, the search distribution must reflect the fitness values of the selected set. This imply that the greater the fitness value of an individual is, the greater the probability of sampling the region around such individual must be. Hence, for the sake of sampling intensively the most promising regions, it is desirable that the probability density function $f(x,t)$ accomplish the following characteristics:

1. Considering the objective/fitness function as $g(x)$, if $g(x_1) > g(x_2)$ then $f(x_1,t) > f(x_2,t)$.
2. When the fitness function $g(x)$ increases/decreases, the probability density function (PDF) increases/decreases.
3. As consequence, if $f(x,t)$ is the PDF, and x^* is the unique maximum, then $f(x^*,t) > f(x,t)$, for any generation t, and $x \neq x^*$.

A widely used and successful way of achieving the mentioned characteristics is by using the Boltzmann distribution, Equation 1, as the conceptual search distribution. Where $g(x)$ is the fitness function. For this article we use the objective function as fitness function directly.

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

$$p(x) := \int_x \frac{\exp\left(\beta \cdot g(x)\right)}{Z} dx \tag{1}$$

The Boltzmann distribution was introduced to explain the distribution of distinguishable particles in several energy states. The Z parameter is a normalization value in order to have a probability function which integrates 1.

Z could be computed by integrating the exponential function in the numerator in the whole search domain, which requires to know the function value for all the points in the search domain, thus it is one of the reasons why we approximate the Boltzmann by a Gaussian. The parameter β has been called the **exponentiation factor** in the EDAs context [16, 10], it is related with the inverse temperature and the Boltzmann constant in the original distribution. Nevertheless, when defining the distribution as in Equation 1, β is related with the selection pressure: if $g(x)$ is the fitness function, the greater fitness function, the greater the probability, additionally, if β is large enough (infinite valued), the optimum has probability 1.

It has been proven that conceptual EDAs based on the Boltzmann selection, such as the Boltzmann EDA (BEDA) [11, 9], converge to the optimum [16]. This work is about an indirect way to use the Boltzmann distribution through the Gaussian model which best explains it, according to the Kullback-Leibler divergence. The goal is to preserve the desired characteristics of the Boltzmann distribution, while maintaining a low computational cost in the estimation and sampling steps.

This approach conceptually revises the EDA goals, by arguing that one of the most important aims in EDAs is to sampling intensively the most promising regions. If this goal is accomplished, then we only will evaluate promising candidate solutions. Otherwise, we could be sampling and evaluating useless candidate solutions. Notice that, frequently, practical EDAs does not accomplish that goal by one or all the following issues: 1) most of the selected could be not positioned in the most promising region. Thus the search distribution could be biased to the region containing most of the selected individuals but not where the most fittest individuals are. 2) If the shape of the search distribution could not capture the selected set structure, then promising regions would not be sampled intensively. These issues have been studied and tackled by researchers, by instance, by inferring which solutions are promising before evaluating [8]. Notice that this way of approaching such issue is a correction step, while our proposal focuses on sampling as many promising solutions as possible, avoiding to sample promising and non-promising solutions and then reject some of them.

The organization of this paper is the following: Section 2 develops the formulae to approximate the Boltzmann PDF with a Gaussian PDF and the computation of the parameters of the Gaussian. Section 3 explains the Boltzman Univariate Distribution Algorithm (BUMDA). Section 4 presents an analysis of the BUMDA characteristics an expected performance. Section

5 provides test problems and performance analysis for comparison with state of the art EDAs. Section 6 presents the main conclusions and discussion about the proposal presented.

2. Approximating the Boltzmann PDF with a Gaussian Model

This Section tackles the approximation of a univariate Boltzmann distribution $P(x) = P_x = \exp(\beta g(x))/Z$, by a univariate Gaussian model $Q(x, \mu, v) = Q_x$. Where the parameters are the mean μ and variance v of the distribution. The Gaussian model for *independent* variables is given by Equation (2).

$$Q(x) = \prod_{i=1}^{n} Q_i(x), \quad where \quad Q_i(x) = Q(x_i, \mu_i, v_i) = \frac{e^{\left[-\frac{(x_i-\mu_i)^2}{2v_i}\right]}}{(2\pi v_i)^{1/2}} \tag{2}$$

A widely used measure of the difference between two distributions $P(x) = P_x$ and $Q(x, \mu, v) = Q_x$ is the Kullback-Leibler divergence (KLD) given in Equation (3). In order to approximate the Gaussian distribution Q_x to the Boltzmann distribution P_x, we minimize the Kullback-Leibler divergence with respect to the Gaussian parameters (μ, v), as shown in Equation (3).

$$K_{Q,P} = \int_x Q_x \log \frac{Q_x}{P_x} dx, \quad \frac{\partial K_{Q,P}}{\partial \theta} = \int_x \left[1 + \log \frac{Q_x}{P_x}\right] \frac{\partial Qx}{\partial \theta} dx. \tag{3}$$

By substituting $\frac{\partial Q_x}{\partial \mu} = Q_x \frac{(x-\mu)}{v}$ into (3) we get Equation (4).

$$\begin{aligned} \frac{\partial K_{Q,P}}{\partial \mu} &= \int_x \left[1 - \frac{(x-\mu)^2}{2v}\right] Q_x \frac{(x-\mu)}{v} dx \\ &- \int_x \left[\log 2\pi v^{1/2} - \log Z + \beta g(x)\right] Q_x \frac{(x-\mu)}{v} dx. \end{aligned} \tag{4}$$

The fact that $(x-\mu)$ is an *odd function* about μ becomes useful to evaluate some integrals, which become equal to 0. We get:

$$\frac{\partial K_{Q,P}}{\partial \mu} = -\frac{\beta}{v} \int_x Q_x (x-\mu) g(x) dx \approx -\frac{\beta}{v} \sum_{x_j \in X} (x_j - \mu) g(x_j). \tag{5}$$

Where X is the selected set. Gallagher and Frean [3] used the gradient approximation in Equation (5) and μ^t to compute μ^{t+1}. In this work we propose to directly compute the μ value which best fits the known information about the fitness function (selected set objective values), as shown in Equation (6):

$$\mu \approx \frac{\sum_j g(x_j)x_j}{\sum_j g(x_j)}. \tag{6}$$

In the same way, substituting $\frac{\partial Q_x}{\partial v} = Q_x\left(\frac{(x-\mu)^2}{2v^2} - \frac{1}{2v}\right)$, into (3):

$$\begin{aligned}\frac{\partial K_{Q,P}}{\partial v} = \int_x \left[1 + \log\frac{Q_x}{P_x}\right] Q_x\left(\frac{(x-\mu)^2}{2v^2} - \frac{1}{2v}\right)dx = \\ \int_x \left[1 + \log[(2\pi v)^{1/2}]\right] Q_x\left[\frac{(x-\mu)^2}{2v^2} - \frac{1}{2v}\right]dx + \\ \int_x \left[-\frac{(x-\mu)^2}{2v} + \log Z - \beta g(x)\right] Q_x\left[\frac{(x-\mu)^2}{2v^2} - \frac{1}{2v}\right]dx\end{aligned} \tag{7}$$

By substituting, in Equation (7), the following equalities:

$$\int_x Q_x(x-\mu)^2dx = v,\ \ \int_x Q_x dx = 1, \text{ and } \int_x (x-\mu)^4 Q_x dx = 3v^2,$$

We obtain Equation (8), which is set equal to 0, in order to minimize the KLD.

$$-\frac{3}{4v} - \frac{\beta}{2v^2}\int_x g(x)Q_x(x-\mu)^2dx + \frac{1}{4v} + \frac{\beta}{2v}\int_x g(x)Q_x dx = 0. \tag{8}$$

Finally, the expressions to analytically compute the variance and its numerical stochastic approximation are given by Equation (9).

$$v = \frac{\int_x g(x)(x-\mu)^2 Q_x dx}{\frac{-1}{\beta} + \int_x g(x)Q_x dx}, \qquad v \approx \frac{\sum_{x_j \in X}\ g(x_j)(x_j-\mu)^2}{T' + \sum_{x_j \in X}\ g(x_j)}, \tag{9}$$

In order to simplify the Equation (9), consider the following:

- $v = \frac{\int_x g(x)(x-\mu)^2 Q_x dx}{\frac{-1}{\beta}+\int_x g(x)Q_x dx} = \frac{\beta \int_x g(x)(x-\mu)^2 Q_x dx}{-1+\beta \int_x g(x)Q_x dx}$. Consequently, for large values of β:

$$\underset{\beta\to\infty}{v} = \frac{\beta \int_x g(x)(x-\mu)^2 Q_x dx}{\beta \int_x g(x)Q_x dx} = \frac{\int_x g(x)(x-\mu)^2 Q_x dx}{\int_x g(x)Q_x dx}. \tag{10}$$

  Also, for large values of the objective function:

$$\underset{g(x)>>0,(x-\mu)^2>0}{v} \approx \frac{\beta \int_x g(x)(x-\mu)^2 Q_x dx}{\beta \int_x g(x)Q_x dx} = \frac{\int_x g(x)(x-\mu)^2 Q_x dx}{\int_x g(x)Q_x dx}. \tag{11}$$

  For these cases:

$$\underset{g(x)>>0,\quad (x-\mu)^2>0 \quad or \quad \beta>>0}{v} \approx \frac{\sum_{x_j\in X} g(x_j)(x_j-\mu)^2}{\sum_{x_j\in X} g(x_j)} \tag{12}$$

- A second consideration is if $\sum_i g(x_i) \approx 0$, for this case there is a numerical problem in the following division: $\frac{\sum_{x_j\in X} g(x_j)(x_j-\mu)^2}{\sum_{x_j\in X} g(x_j)}$.

- A third consideration for the Equation: $\frac{\sum_{x_j\in X} g(x_j)(x_j-\mu)^2}{\sum_{x_j\in X} g(x_j)+T'}$ is: if $T' < 0$ and $\sum_{x_j\in X} g(x_j) < |T'|$, then $v < 0$ (considering that $g(x) > 0 \forall x$).

According to these considerations:

- T' must be greater than 0.

- The value of T' becomes irrelevant, for large values of β or large values of $g(x)$. Even more, β itself becomes irrelevant, according to Equations 10,11, 11.

Due to these considerations we propose a $T' = 1$. That means that we assume a *sufficiently large* β value, and we avoid numerical problems. Notice that a small beta value could be used for regulating the variance, only for increasing the variance, it can not be decreased by β.

This section concludes with two important result given by Equations (6) and (9), which are the needed formulae to compute the parameters of a univariate Gaussian model which approximate the Boltzmann distribution by minimizing the KLD. A possible drawback of the univariate model is that it is

restricted to problems which present weak variable correlation. On the other hand, the advantages of this model are simplicity and low computational cost, not to mention the promising results reported, such as the UMDA$_c^G$ [6], PBIL [1] and BG-UMDA [15].

3. The Boltzmann Estimation of Distribution Algorithm (BUMDA)

Two desired characteristics of an EDA, and in general of any evolutionary algorithm, are the following:

- A non-decreasing sequence of the expected value of the population fitness function. In order to obtain better samples than the generation before.
- Convergence to the best solution found. In order to refine the solution, and to determine when the algorithm rarely will improve the best solution known.

A simple way to ensure both characteristics is to apply a truncation selection method which increases the mean of the fitness value, such as explained in Figure 1. As the mean of the fitness value of the elected set (and the population) is bounded by the elite fitness value, then, the mean converges to it. We ensure that it is always at least one element in the selected set by preserving the elite individual.

Truncation Selection Method

Consider a population of decreasingly sorted individuals (maximization case), such that x_1 are the decision variables of the individual with the maximum objective function in the population. :

1. For the initial generation $t = 0$, let be $g(x_i, 0)$ for $i = 1..N$, the objective values of the initial population. Define: $\theta_0 = \min g(x_i, 0)$.
2. For $t > 0$, set:
   $\theta_t = \max\big(g(x_{N/2}, t), \min(g(x_i, t)|g(x_i, t) \geq \theta_{t-1})\big)$.
3. Truncate the population such that $g(x_s, t) \geq \theta_t$. Where x_s are all the individuals whose objective values are equal or greater than θ_t.

Figure 1: Truncation method to ensure convergence in a population based algorithm.

Now, we have all the elements needed to introduce the BUMDA, shown in Figure 2. Notice that the BUMDA uses the truncation selection method to ensure an increasing average (mean estimator) of the objective function of the population. In addition, the fitness function of the selected set is used to incorporate information about the fitness landscape into the Gaussian model.

A simplification for the variance calculation was done by setting $T' = 1$. A reason to set $T' = 1$ is due to analysis of several experiments conducted, which suggest that the BUMDA performance is significantly more impacted by changes in the population size than the value of T'.

The reader must observe that the fixed T' does not imply a fixed distribution because the distribution is computed according the selected set which is changing every generation. Hence, the current distribution discard all the regions in which $g(x) < \theta$, as it is shown in Figure 3.

BUMDA

1. Give the parameter and stopping criterion:
   nsample $\leftarrow$ Number of individuals to be sample.
   minvar $\leftarrow$ minimum variance allowed.
2. Uniformly generate the initial population P_0, set $t = 0$.
3. While $v > minvar$ for all dimensions
   (a) $t \leftarrow t + 1$
   (b) Evaluate and truncate the population according algorithm in Figure 1.
   (c) Compute the approximation to μ and v (for all dimensions) by using the selected set (of size $nselec$), and Equations (6) and (9), as follows:

   $$\mu \approx \frac{\sum_1^{nselec} x_i \bar{g}(x_i)}{\sum_1^{nselec} \bar{g}(x_i)}, \quad v \approx \frac{\sum_1^{nselec} \bar{g}(x_i)(x_i - \mu)^2}{1 + \sum_1^{nselec} \bar{g}(x_i)},$$

   where $\bar{g}(x_i) = g(x) - g(x_{nselec}) + 1$.
   Note: the individuals can be sorted to simplify the computation, and $g(x_{nselec})$ is the minimum (for maximization case) objective value of the selected individuals.
   (d) Generate $nsample - 1$ individuals from the new model $Q(x, t)$, and insert the elite individual.
4. Return the elite individual as the best approximation to the optimum.

Figure 2: Pseudo-code for BUMDA

According to the proposals presented in this section we infer some interesting characteristics, which will be discussed in the next section:

- The BUMDA converges to the best approximation to the optimum.

- The variance tends to 0 for a large number of generations.
- The BUMDA only needs **one** parameter (population size).
- The estimation of the search distribution parameters results in a fast automatic adaptation. The variance could be increased or decreased, according to the solutions in the selected set and their objective values, and the mean moves fast to the region where the best solutions are.

4. BUMDA Analysis

This section presents a brief analysis of the BUMDA characteristics mentioned at Section 3. Firstly we analyze the convergence property of BUMDA, and in general the convergence with the truncation method shown in Figure 1. Secondly we discuss the tendency of variance to 0 for a large number of generations, which can be used as a stopping criterion. Finally, we present the differences between BUMDA and maximum likelihood estimation.

4.1. BUMDA Convergence

Let us call the worst objective value at the initial population as θ_0, and the best objective value found by BUMDA during all the generations as θ_n. Due to step 2 in Figure 1, $\theta_t \geq \theta_{t-1}$, thus the set $\{\theta_t\}$ is a non-decreasing sequence. Also note that the θ_t value is always taken from the population generated during the search process, and the best value generated through all the generations is θ_n (the last objective value of the elite individual). Then we have a non-decreasing sequence upper bounded by θ_n. Note that the probability of sampling an individual x_θ with the same value of θ_{t-1} is 0, say $P(g(x_\theta) = \theta_{t-1}) = 0$, by consequence $P(\theta_t > \theta_{t-1}) = 1$, then the non-decreasing sequence $\{\theta_t\}$ becomes an strictly increasing sequence. Note that the objective values of the selected set are always greater or equal to θ_t, say $g(x_s) \geq \theta_t$, then the whole selected set converge to θ_n. That is to say, for any continuous function, all the points will be clustered around the best solution found. The convergence of the whole population to a point is especially important in order to use a variance measure as stopping criterion. The Figure 3 graphically shows the effect of the truncation method during several generations.

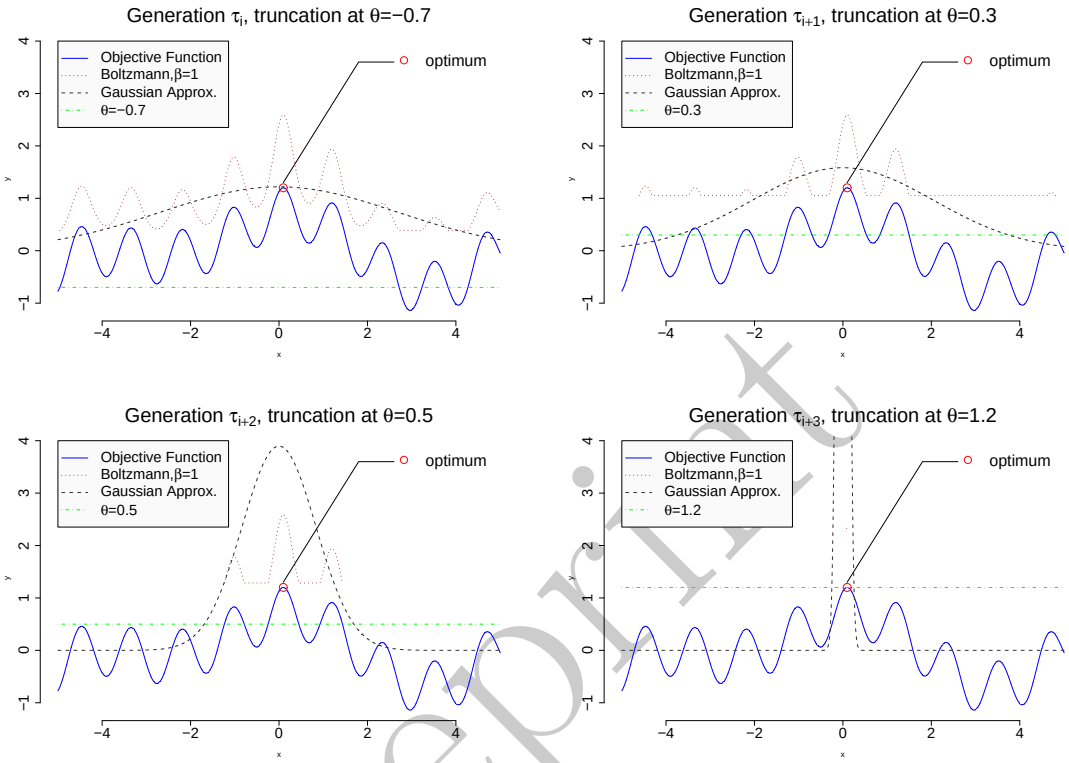


Figure 3: The effects of the truncation method for the BUMDA.

4.2. BUMDA parameter adaptation

It has been noticed that in general the maximum likelihood (ML) parameter estimation is not the best strategy for approximating the optimum in an EDA [5]. Some alternative strategies have been proposed [4]. This section show how the BUMDA parameter estimation differs from ML estimation, and how these differences improve the search process.

Consider a selected set as the one shown in Figure 4 (labeled as sample points). This selected set is used to compute the parameters for the Gaussian distribution. Suppose that most of the population has been clustered around $x = -1$, and new promising solutions have been discovered near to the optimum around $x = 6$. The dashed line is the density function obtained when using the ML formulae, the bold line is the density function obtained by using the BUMDA formulae.

The weights used in the BUMDA leads the Gaussian mean (vertical bold

line) closer to the optimum than the ML mean (vertical dashed line), additionally the variance for the BUMDA density is larger than the ML variance, thus, BUMDA increases its exploration capacity in a region closer to the optimum.

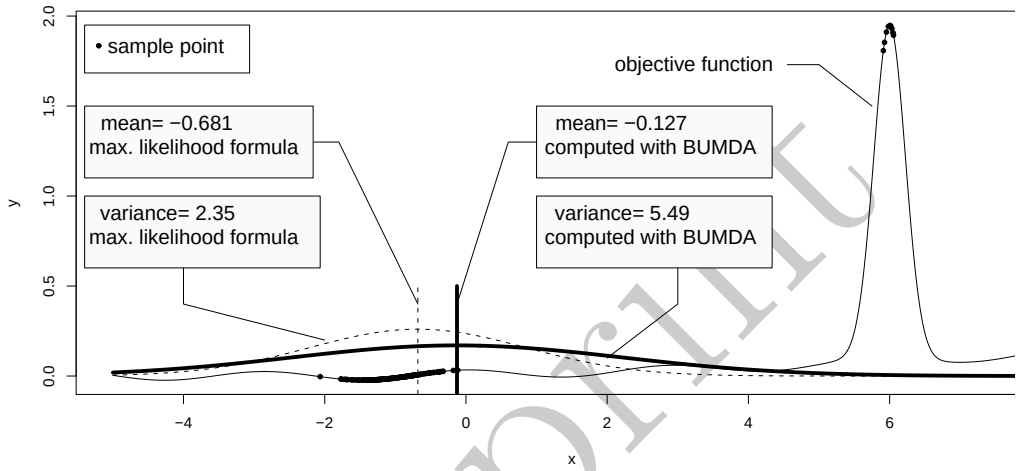
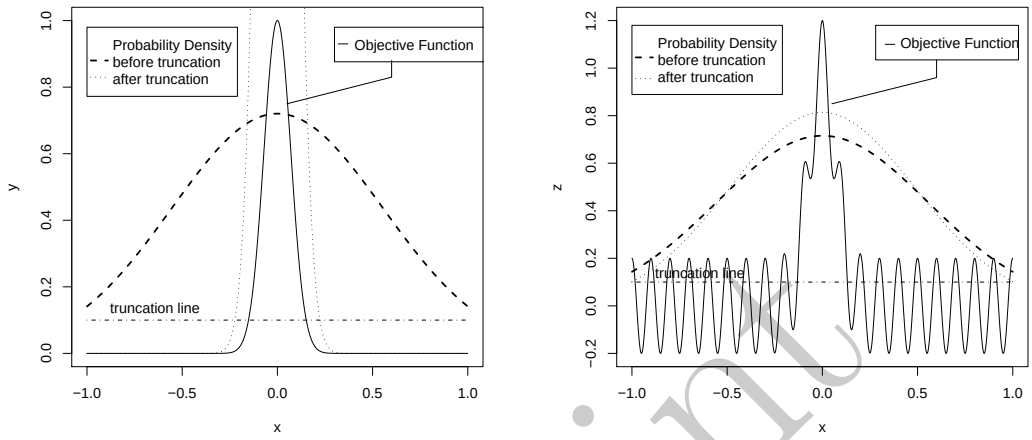


Figure 4: Comparison between BUMDA and maximum likelihood parameter estimation.

In the same vein, the BUMDA implicitly incorporate information about the multiple local maxima in the objective function. This characteristic is shown in Figure 5. An equally spaced sample has been taken in the same domain in functions (a) and (b), then we compute the mean and variance according to BUMDA formulae. Notice that when optimizing the function in Figure 5(a) the distribution has a smaller variance than that in Figure 5(b), even though both samples use the same set of points before truncation. When the population is truncated the BUMDA detects that a wider exploration is needed for the function with more local maxima. The mentioned characteristics of the BUMDA, justifies the application of both truncation and Boltzmann selections, the first helps to achieve convergence and the latter incorporates information of the function landscape.

5. Test Problems and Performance Analysis

This section presents experiments and comparison among the BUMDA and state of the art EDAs proposed by different researchers. The BUMDA



(a)
$\mu_{bf} = 0, v_{bf} = 0.306$
$\mu_{af} = 0, v_{af} = 0.006$

(b)
$\mu_{bf} = 0, v_{bf} = 0.311$
$\mu_{af} = 0, v_{af} = 0.240$

Figure 5: Comparison of functions with different local maxima. We took equally spaced points in both cases a) and b), both samples have the same average in x and y, but note how the function with more local maxima has a larger variance. Thus, the BUMDA maintains a wider exploration when the sample is truncated. μ =mean, v =, variance, bf = before truncation, af = after truncation.

is compared with an univariate state of the art EDA, for instance, BG-UMDA [15] which is the most similar approach in the literature. In addition, the BUMDA is compared with multivariate Gaussian based EDAs such as the EMNA-B [15], the Iterated Density Estimation Evolutionary Algorithm (IDEA), the Evolution Strategy with Covariance Matrix Adaptation (CMA-ES), and the Correlation-Triggered Adaptive Variance Scaling IDEA (CT-AVS-IDEA) [4]. This multivariate EDAs can be unified as multivariate Gaussian based estimation of distribution algorithms [2], which have in common a similar complexity in the model. This kind of algorithm intends to capture the search directions from the structure of the selected set. Our experiments show that an adequate bias of the search distribution via the selection step, is at least as important as the model used to capture the selected set structure.

5.1. Problem Test 1.

This set of problems, taken from [14], compares the BUMDA with the EMNA-B and BG-UMDA reported in [15]. . The BG-UMDA also uses an univariate Gaussian function to approximate the Boltzmann distribution. This set of functions has been widely used to compare EDAs [7]. Some of these functions have many local maxima/minima. In addition, as the functions are defined for any dimension, this set could be used to analyze the scalability of the algorithms.

Experiment and BUMDA parameter settings. All the algorithms were tested for 3×10^5 function evaluations or when they found a solution with an error less than or equal to 10^{-6}. To make a fair comparison we used the same stopping criteria of the experiments reported in [15]. The population size for this test is 3000 for the Sum Cancellation, and 300 for all the other functions. In general, the BUMDA population size could be set in a straightforward way, increasing it until the best optimum approximation is found or the performance does not change.

Results analysis. The BUMDA finds the best average value of the objective functions in three of five cases, as shown in Table 1, and it is significantly better than BG-UMDA in two cases. On the other hand, as can be noticed the BUMDA fails to reach a close optimum approximation in the Sum Cancellation and Rosenbrock problems. These functions have a multivariate interaction, hence it is reasonable that the best performing algorithm is the EMNA-B. When looking at the objective function value, we can conclude that the BUMDA is competitive in such problems which do not require a multivariate model. In other vein, since the approximation error is used as stopping criterion, a useful comparison is given by the number of function evaluations required to reach it. See Table 2, the BUMDA uses the less average number of evaluations for those cases which are successfully solved by it. Observe that the difference in the number of evaluations between 10 dimension and 50 dimension problems increased less than 3 times but the dimensionality increased 5 times (when the optimum is found by BUMDA). According to the results just presented, we can derive various observations:

- The BUMDA performs well in univariate problems, as expected.
- The selection pressure given by the BUMDA selection method: truncation/weighting, is more adequate than such of the BG-UMDA for this kind of problems, according to Table 2, because with a smaller number

of function evaluations the BUMDA delivers similar or better results than the BG-UMDA.

- As can be notice, the Sphere and Griewangk functions require a similar computational effort to reach the desired optimum approximation, while the Ackley function has a different requirement. The explanation is that the Griewangk function in high dimensions (10 or more) becomes similar to the sphere. The three compared algorithms require a similar number of function evaluations in the Griewangk and Sphere functions for 50-dimension, and quite similar also for the 10-dimension case. As can be noticed the number of function evaluations depends completely on the problem, similar problems have a similar cost. This means that the three algorithms are using information about the function landscape to perform the search. This is interesting because instead of require as user parameter the number of generations or evaluations, one can fix the desired precision, by using a variance based stopping criterion which is a more easy-to-tune parameter, because usually it completely depends on the optimization problem.

Statistical test. The comparison among the BUMDA, the BG-UMDA and EMNA-B uses the $z - test$ with $\alpha = 0.05$. It is used to compare both, objective values and function evaluations. This is the recommendable test because the only data available are the means and standard deviations. The $t - test$ should not be used, because in general the variances could not be considered homogeneous, according with the F_{max} test. The rightmost column of Tables 1 and 2 show the $z - test$ results. If the alternative hypothesis H_1 is accepted, the BUMDA is better than the other algorithm. Otherwise the null hypothesis is not rejected, therefore, there is not enough statistical evidence to say which algorithm is better.

5.2. Problem test 2

This set of problems is taken from [4]. All the functions are convex and have been generalized for any number of dimensions. Most of these problems can be solved by well performed EDAs as the presented in [4]. Then, an objective comparison, must be based on scalability or effort needed by the algorithm to reach the optimum, by relating the number of evaluations with the problem dimensionality. For these problems we report a plot of the problem dimensionality (2,4,8,10,20,40,80) versus the average number of

Function	**BUMDA**	EMNA-B	BG-UMDA	$H_1 : \bar{F}_{BUMDA}$ *better than* $\bar{F}_{other}$ EMNA-B	BG-UMDA
SumC 10d	$7.5E3 \pm 8.4E3$	$1E5 \pm 1.1E$-7	$5.8E4 \pm 2.3E4$	no	no
SumC 50d	2.07 ± 0.12	99910 ± 160	1.39 ± 0.1	no	yes
Grie. 10d	**7.3E-7 ± 1.7E-7**	$7.4E$-7 $\pm 1.1E$-7	$1.27E$-4 $\pm 4E$-4	no	no
Grie. 50d	**9E-7 ± 8.4E-8**	$9.2E$-7 $\pm 5E$-8	$8.8E$-7 $\pm 7E$-8	no	no
Sphe. 10d	**7E-7 ± 1.6E-7**	$7.5E$-7 $\pm 2.1E$-7	$5.9E$-7 $\pm 1.8E$-7	no	no
Sphe. 50d	**8.7E-7 ± 8.1E-8**	$8.8E$-7 $\pm 1.1E$-7	$8.4E$-7 $\pm 8E$-8	no	no
Rose. 10d	8.1 ± 0.08	6.33 ± 0.37	7.74 ± 0.08	no	no
Rose. 50d	47.7 ± 0.18	47.08 ± 0.44	47.54 ± 0.07	no	no
Ackl. 10d	**8.3E-7 ± 1.2E-7**	$8.4E$-7 $\pm 1E$-7	$8.3E$-7 $\pm 1.6E$-7	no	no
Ackl. 50d	**9.3E-7 ± 4.3E-8**	$9.42E$-7 $\pm 4E$-8	$9.6E$-7 $\pm 4E$-8	no	yes

Table 1: Mean and standard deviation of best function value found in 20 runs for the Test problem 1. **yes= the BUMDA is better than the other algorithm.**

Function	**BUMDA**	EMNA-B	BG-UMDA	$H_1 : \bar{N}^{eval}_{BUMDA} < \bar{N}^{eval}_{other}$ EMNA-B	BG-UMDA
SumC. 10d	$3E5 \pm 0$	92520 ± 840	300400 ± 0	NP	NP
SumC. 50d	$3E5 \pm 0$	301000 ± 0	300400 ± 0	NP	NP
Grie. 10d	**17262 ± 384**	134000 ± 47000	$229E3 \pm 64E3$	yes, p=5.8E-29	yes, p=7.8E-50
Grie. 50d	**39675 ± 342**	170100 ± 1700	71880 ± 420	yes, p=0	yes, p=0
Sphe. 10d	**14541 ± 261**	35200 ± 420	35720 ± 840	yes, p=0	yes, p=0
Sphe. 50d	**40695 ± 325**	192900 ± 1600	82400 ± 460	yes, p=0	yes, p=0
Rose. 10d	$3E5 \pm 0$	300400 ± 0	300400 ± 0	NP	NP
Rose. 50d	$3E5 \pm 0$	301000 ± 0	300400 ± 0	NP	NP
Ackl. 10d	**23257 ± 287**	43560 ± 610	44000 ± 530	yes, p=0	yes, p=0
Ackl. 50d	**58850 ± 348**	231800 ± 4300	98920 ± 530	yes, p=0	yes, p=0

Table 2: Average and standard deviation of evaluations for Test problem 1. **yes= the BUMDA is better than the other algorithm. NP= Comparison Not Possible.**

evaluations (to preserve the experimental conditions of the results presented in [4]), as well as a regression coefficient.

The comparison includes well performed algorithms reviewed in [4]: the Iterated Density Estimation Evolutionary Algorithm (IDEA), the Evolution Strategy with Covariance Matrix Adaptation (CMA-ES), and the Correlation-Triggered Adaptive Variance Scaling IDEA (CT-AVS-IDEA). The BUMDA successfully solves 30 independent consecutive runs for all the test problems except the Rosenbrock (which is not presented). The linear least squares regressions on log-log data are presented in Table 3, where the average number of evaluations e depends on the dimensionality l as follows:

$$\log e = \beta \log l + \beta_0 + \epsilon \tag{13}$$

The regression coefficient β can be seen as an empirical order of the algorithm, it shows how the number of evaluations grows with the dimensionality.

Experiment and BUMDA Parameter Setting. All the algorithms use the closeness to the optimum as termination criterion, it was set in 10^{-10} for all the functions except the different powers function which optimum closeness was set in 10^{-15}. The population size for the sphere is 200, for the ellipsoid is 200 from 2 to 10 variables, 400 for 20 and 40 variables, and 500 for 80 variables. For the cigar function, 200 for 2 and 4 variables, 300 for 8 and 10 variables, 400 for 20, 600 for 40 and 900 for 80. For the tablet, 200 from 2 to 20 variables, 300 for 40, and 400 for 80. For the two axes 200 from 2 to 8 variables, 300 for 10, 500 for 20, 600 for 40, and 700 for 80. For the different powers 200 from 2 to 8, 300 for 10, 600 for 20, 1200 for 40 and 2400 for 80. For the parabolic ridge 300 from 2 to 10 variables, 400 for 20, 500 for 40, and 600 for 80. For the sharp ridge 200 from 2 to 20 variables, 300 for 40 and 400 for 80.

Results analysis. As shown in Table 3 the BUMDA order can be considered less than the other EDAs compared, even more, most of the problems have a regression coefficient closed to 1, this means linear scaling. This is quite important, in spite of the fact that the objective function used in this test are all convex, the BUMDA performs better than other algorithms which niche of application are convex functions. The results in Table 3 shows that for this kind of problems the BUMDA computational scalability is sublinear or linear at maximum.

The BUMDA plot in Figure 6 shows the linear behavior, of BUMDA, the reader can compare this plot with the presented in [4], in order to observe how BUMDA can outperform more complex models such as multivariate Gaussians. The symbols used to represented the different test problems are: **Cigar + , Cigar tablet $\times$, Different powers *, Ellipsoid $\square$, Parabolic Ridge $\blacksquare$, Sharp Ridge $\bullet$, Sphere $\triangle$, Tablet $\blacktriangle$, Two axes $\triangledown$.**

6. Conclusions

According to the results obtained it is worth to notice that the BUMDA represents a different point of view in EDAs: while many researchers have presented proposals which intend to capture as better as possible the fitness landscape in a probability model, using very complex and computational expensive models [13] [17], the BUMDA shows that complex probability mod-

Function	IDEA	AVS-IDEA	CMA-ES	BUMDA
Sphere	1.1635	1.6563	0.9601	**0.6250**
Ellipsoid	1.2171	1.6870	1.1093	**0.9214**
Cigar	1.1865	1.6976	1.1093	**1.0817**
Tablet	1.0860	1.6397	1.4178	**0.7679**
Cigar Tablet	1.1142	1.7155	1.2431	**1.03823**
Two Axes	1.2854	1.6551	1.7208	**1.0437**
Different Powers	1.4983	1.1692	1.5845	**1.3487**
Parabolic Ridge	not solved	1.1160	1.0853	**0.7956**
Sharp Ridge	not solved	0.8563	1.4764	**0.7959**
Worst Coefficient	1.4983	1.9154	1.7208	**1.3487**
Best Coefficient	1.086	0.8563	0.9601	**0.6250**
Coefficient Average	1.2216	1.5108	1.3603	**0.9353**

Table 3: Regression coefficients can be seen as an empirical order of the algorithm, thus in average the BUMDA is $O(n^{0.9353})$ (sublinear), where n is the number of variables.

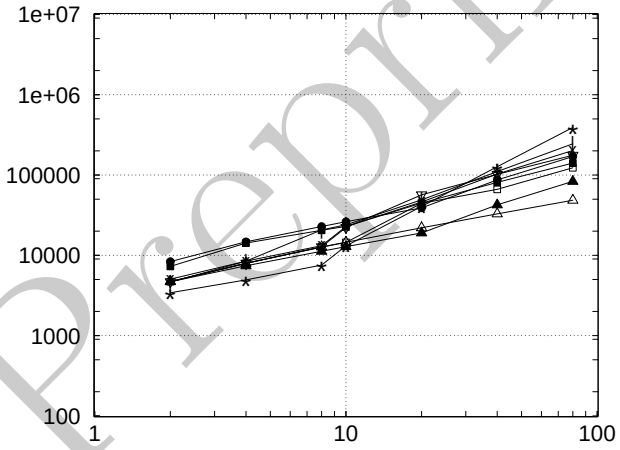


Figure 6: Dimensionality (X-axis) vs Average Number of Evaluations (Y-axis), plotted in log-log scale for 30 independent runs of BUMDA.

els can be outperformed by a simple univariate Gaussian distribution. The explanation is the focus on increasing the probability of finding the optimum (sampling intensively promising regions) instead of fitting the data. In this vein, when using maximum likelihood estimation or any other estimator which fits the data, there is not guaranty that the most promising regions will be intensively sampled, because: 1) The data could have a wrong bias. It is to say, we have many individuals with a middle-range fitness value and

few *fittest individuals* in the selected set, as they are usually equally weighted for the parameter computation, the most promising regions could be lost. 2) The search distribution model is incapable of capturing the structure of the data. For example, if the *fittest individuals* are clustered in two separated regions, and the search distribution uses a unimodal model, it is very possible that most of the probability mass be positioned in the middle of the most promising regions, instead of one of them. On the other hand, the BUMDA actually intends to fit a Gaussian which has the maximum probability value in the best known solution. The last statement does not mean that complex models or distribution factorizations are useless, but that the parameter computation must ensure that the most promising regions are intensively explored, regardless the parametric model used as search distribution.

The Boltzmann Univariate Marginal Distribution Algorithm (BUMDA), ensures convergence to the best solution found for a large number of (increasing-expectation) generations. The BUMDA can solve an extensive type of problems with a very competitive effort (number of evaluations). In addition, the computational cost required to calculate the parameters of the probabilistic model is $O(nm)$ (linear) with the number of dimensions n, and the population size m. The order of the algorithm empirically computed, shows that the function evaluations grow sublinearly $O(n^{0.9353})$ with respect to the number of variables. The BUMDA achieves the reduction of user-given parameters, requiring just ***one***: the population size, which can be easily tuned.

We suggest to use a stopping criterion based on minimum variance, because this criterion detects a poor exploration and when the optimum approximation is being rarely improved.

The Test problems presented in Section 5, are used to contrast convergence, optimum approximation and scalability of the BUMDA versus state of the art EDAs. The results provide evidence about the BUMDA competitiveness when it is compared with approaches based on univariate models, such as BG-UMDA, presented in Test problems 1. Even more, the BUMDA is competitive in the accuracy of the optimum approximation and the computational cost with multivariate models such as: EMNA-B, IDEA, CT-AVS-IDEA and CMA-ES, presented in Test problems 1 and 2.

Future work will contemplate the approximation of the Boltzmann distribution by a more complex model which captures dependencies among variables. Another important issue is to adopt the concept of using a search distribution which really incorporates information about the fitness landscape, and allows to sample more intensively the most promising regions.

Finally, according to our results we conclude that new EDAs proposals must consider the following issues: 1) to use all the information at hand to perform the exploration, particularly to use the fitness values of the population to estimate the search distribution, and 2) to use the explicit probabilistic modeling in EDAs to ensure that the most promising regions (the regions known with the best objective function), be intensively explored. 3) to use the explicit probabilistic modeling in EDAs to explain: when EDAs should work, why EDAs should work, and which guidelines one must follow in order to design successful EDAs.

References

[1] S. Baluja, Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning, Tech. rep., Carnegie Mellon University, Pittsburgh, PA, USA (1994).

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

[3] M. Gallagher, M. Frean, Population-based continuous optimization, probabilistic modelling and mean shift, Evol. Comput. 13 (1) (2005) 29–42.

[4] J. Grahl, P. A. 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, New York, NY, USA, 2006, pp. 397–404.

[5] J. Grahl, P. A. N. Bosman, S. Minner, Convergence phases, variance trajectories, and runtime analysis of continuos EDAs, in: GECCO '07: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM, 2007, pp. 516–522.

[6] P. Larrañaga, R. Etxeberria, J. Lozano, J. Peña, Optimization by learning and simulation of Bayesian and Gaussian networks, Tech. Rep. EHU-KZAA-IK-4/99, Department of Computer Science and Artificial Intelligence, University of the Basque Country (1999).

[7] P. Larrañaga, J. A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[8] H. Luong, H. Nguyen, C. W. Ahn, Entropy-based efficient enhancement techniques for evolutionary algorithms, Information Sciences 188 (2012) 100–120.

[9] T. Mahnig, H. Mühlenbein, Comparing the adaptive boltzmann selection schedule SDS to truncation selection, in: Proceedings of the Third International Symposium on Adaptive Systems ISAS 2001, Evolutionary Computation and Probabilistic Graphical Models, La Habana, Cuba, 2001, pp. 121–128.

[10] H. Mühlenbein, Convergence theorems of estimation of distribution algorithms, Markov Networks in Evolutionary Computation (2012) 91–108.

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

[12] H. Mühlenbein, G. Paaβ, From recombination of genes to the estimation of distributions i. binary parameters, in: PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, Springer-Verlag, London, UK, 1996, pp. 178–187.

[13] R. Salinas-Gutiérrez, A. Hernández-Aguirre, E. R. Villa-Diharce, Dependence trees with copula selection for continuous estimation of distribution algorithms, in: Proceedings of the 13th annual conference on Genetic and evolutionary computation, GECCO '11, ACM, New York, NY, USA, 2011, pp. 585–592.

[14] M. Sebag, A. Ducoulombier, Extending population-based incremental learning to continuous search spaces, in: Parallel Problem Solving from NaturePPSN V, Springer, 1998, pp. 418–427.

[15] C. Yunpeng, S. Xiaomin, J. Peifa, Probabilistic modeling for continuous EDA with Boltzmann selection and Kullback-Leibeler divergence, in: GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation, ACM, New York, NY, USA, 2006, pp. 389–396.

[16] Q. Zhang, H. Mühlenbein, On the convergence of a class of estimation of distribution algorithms, IEEE Transactions on Evolutionary Computation 8 (2) (2004) 127–136.

[17] J.-h. Zhong, J. Zhang, Z. Fan, MP-EDA: a robust estimation of distribution algorithm with multiple probabilistic models for global continuous optimization, in: Proceedings of the 8th international conference on Simulated evolution and learning, SEAL'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 85–94.