Designing EDAs by using the Elitist Convergent EDA Concept and the Boltzmann Distribution.

S. Ivvan Valdez Peña, Arturo Hernández Aguirre, and Salvador Botello Rionda
Dept. of Computer Science, Centre for Research in Mathematics (CIMAT)
C. Jalisco S/N, Guanajuato, Guanajuato, A.P. 402, México
ivvan@cimat.mx, artha@cimat.mx,botello@cimat.mx

ABSTRACT

This paper presents a theoretical definition for designing EDAs called Elitist Convergent Estimation of Distribution Algorithm (ECEDA), and a practical implementation: the Boltzmann Univariate Marginal Distribution Algorithm (BUMDA). This proposal computes a Gaussian model which approximates a Boltzmann distribution via the minimization of the Kullback Leibler divergence. The resulting approach needs only one parameter: the population size. A set of problems is presented to show advantages and comparative performance of this approach with state of the art continuous EDAs.

Categories and Subject Descriptors: I.2[ARTIFICIAL INTELLIGENCE]Miscellaneous

General Terms: Algorithms, Design, Experimentation, Performance, Theory.

Keywords: Estimation of distribution algorithms, Boltzmann distribution, Kullback-Leibler divergence, Performance analysis.

Track: Estimation of Distribution Algorithms.

1. INTRODUCTION

An open question that remain unanswered about Estimation of Distribution Algorithms (EDAs), is: When will an EDA perform successfully? This work presents a general condition for convergence of conceptual EDAs to the optimum, and shows how this concept can be used to design practical algorithms.

Without loss of generality consider a maximization problem. In order to find better solutions at each generation, a logical requirement for an EDA is to increase the expectation of the objective function. This is established in Definition 1.1.

DEFINITION 1.1. *Consider an objective function $g(x)$, a density function $f(x)$, and sequences of consecutive generations $t = 1, 2, 3..N$, and non-consecutive generations $\tau = \tau_1, \tau_2 ... \tau_M$. An Estimation of Distribution Algorithm which fulfills:*

$$\int_X g(x) f(x,t) dx \leq \int_X g(x) f(x, t+1) dx \tag{1}$$

and

$$\int_X g(x) f(x,\tau_i) dx < \int_X g(x) f(x, \tau_{i+1}) dx \tag{2}$$

For all $t \in \mathbf{N}$ and $\tau_i < \tau_{i+1} \in \mathbf{N}$, is called an **Elitist Convergent EDA (ECEDA).**

Consider a sequence $\tau = \tau_1, \tau_2, ..., \tau_M$. It can be proved that an *Elitist Convergent* EDA fulfills that:

$$\lim_{M\to\infty} E(g(x), \tau_M) = max\ g(x) \tag{3}$$

We can write a Boltzmann ECEDA by substituting the Boltzmann probability density function into Equation 1. This work proposes to approximate the Boltzmann by a Gaussian distribution through the minimization of the Kullback-Leibler divergence. The resulting proposal is called Boltzmann Univariate Marginal Distribution Algorithm (BUMDA).

2. THE BOLTZMANN UNIVARIATE MARGINAL DISTRIBUTION ALGORITHM

A simple way to ensure convergence is to apply a truncation method which increases the mean of the population, such as explained in algorithm in Figure 1. Now we can introduce the The Boltzmann Univariate Marginal Distribution Algorithm (BUMDA) in Figure 2, which uses the mean and variance which minimize the Kullback-Leibler divergence between the Boltzmann and Gaussian univariate distributions (using a fixed temperature).

We must ensure that there is always at least one element in the selected set by preserving the elite individual.

3. TEST PROBLEMS AND PERFORMANCE ANALYSIS

The test presented is a general comparison, they are multimodal functions, and convex functions (sphere) to test convergence speed, and they are solved in different dimensions (10 and 50). The comparison for this set is perform among BUMDA, the best performed EDA in other comparison (EMNA-B, [8]), and the BG-UMDA, a similar approach which uses an univariate normal distribution and the Boltzmann function. The BUMDA is the most competitive approach for three problems of this set, as shown in Table 1. The BUMDA

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

Truncation Method

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

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

BUMDA

1. Give the parameter and stop criterion:
   nsample ← Number of individuals to be sample.
   minvar ← 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 $nselec$), as follows:
   $\mu \approx \frac{\sum_1^{nselec} x_i \bar{f}(x_i)}{\sum_1^{nselec} x_i \bar{f}(x_i)}$,
   $v \approx \frac{\sum_1^{nselec} \bar{f}(x_i)(x_i-\mu)^2}{1+\sum_1^{nselec} \bar{f}(x_i)}$,
   where $\bar{f}(x_i) = f(x) - f(x_{nselec}) + 1$.
   Note: the individuals can be sorted to simplify the computation, and $f(x_{nselec})$ is the minimum 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

finds the best average value of the objective functions in most of the cases, but as we are using the solution error as stopping criterion, the real comparison is given by the number of function evaluations. The BUMDA uses the less average number of evaluations for all cases when it finds the solution. Observe that there is not a great difference between the number of evaluations for 10 dimensions and 50 dimensions. Even though the dimensionality was increased 5 times, the number of evaluations increased less than 3 times (when the optimum was found by BUMDA).

Stopping Criteria. All the algorithms were tested for 3×10^5 function evaluations or when they found a solution with an error less or equal to 10^{-6}.

BUMDA Parameter Setting. The population sizes for this test are 3000 for the Sum Cancellation, and 300 for all the other functions.

Function	**BUMDA**	EMNA-B	BG-UMDA
SumC 10d	$7.5E3 \pm 8.4E3$	$1E5 \pm 1.1E\text{-}7$	$5.8E4 \pm 2.3E4$
SumC 50d	2.07 ± 0.12	99910 ± 160	1.39 ± 0.1
Griew. 10d	**7.3E-7 ± 1.7E-7**	$7.4E\text{-}7 \pm 1.1E\text{-}7$	$1.27E\text{-}4 \pm 4E\text{-}4$
Griew. 50d	**9E-7 ± 8.4E-8**	$9.2E\text{-}7 \pm 5E\text{-}8$	$8.8E\text{-}7 \pm 7E\text{-}8$
Sphere 10d	**7E-7 ± 1.6E-7**	$7.5E\text{-}7 \pm 2.1E\text{-}7$	$5.9E\text{-}7 \pm 1.8E\text{-}7$
Sphere 50d	**8.7E-7 ± 8.1E-8**	$8.8E\text{-}7 \pm 1.1E\text{-}7$	$8.4E\text{-}7 \pm 8E\text{-}8$
Rosen. 10d	8.1 ± 0.08	6.33 ± 0.37	7.74 ± 0.08
Rosen. 50d	47.7 ± 0.18	47.08 ± 0.44	47.54 ± 0.07
Ackley 10d	**8.3E-7 ± 1.2E-7**	$8.4E\text{-}7 \pm 1E\text{-}7$	$8.3E\text{-}7 \pm 1.6E\text{-}7$
Ackley 50d	**9.3E-7 ± 4.3E-8**	$9.42E\text{-}7 \pm 4E\text{-}8$	$9.6E\text{-}7 \pm 4E\text{-}8$

Table 1: Mean and standard deviation of best function value found in 20 runs.

Function	**BUMDA**	EMNA-B	BG-UMDA
SumCan 10d	$3E5 \pm 0$	92520 ± 840	300400 ± 0
SumCan 50d	$3E5 \pm 0$	301000 ± 0	300400 ± 0
Griewangk 10d	**17262 ± 384**	134000 ± 47000	$229E3 \pm 64E3$
Griewangk 50d	**39675 ± 342**	170100 ± 1700	71880 ± 420
Sphere 10d	**14541 ± 261**	35200 ± 420	35720 ± 840
Sphere 50d	**40695 ± 325**	192900 ± 1600	82400 ± 460
Rosenbrock 10d	$3E5 \pm 0$	300400 ± 0	300400 ± 0
Rosenbrock 50d	$3E5 \pm 0$	301000 ± 0	300400 ± 0
Ackley 10d	**23257 ± 287**	43560 ± 610	44000 ± 530
Ackley 50d	**58850 ± 348**	231800 ± 4300	98920 ± 530

Table 2: Average and standard deviation of evaluations.

4. REFERENCES

[1] P. A. N. Bosman and D. Thierens. Expanding from discrete to continuous estimation of distribution algorithms: The idea. In *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pages 767–776, London, UK, 2000. Springer-Verlag.

[2] M. Gallagher[1] and M. Frean[2]. Population-based continuous optimization and probabilistic modelling. Technical report, [1]School of computer Science and Electrical Engineering, Univerity of Queensland, Australia. [2]School of Mathematical and Computing Sciences, Victoria Univerty, New Zealand., University of Queensland. 4072 Australia., 2001.

[3] J. Grahl, P. A. N. Bosman, and 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*, pages 516–522. ACM, 2007.

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

[5] T. Mahnig and 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*, pages 121–128, La Habana, Cuba, 2001.

[6] H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3):303–346, 1997.

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

[8] C. Yunpeng, S. Xiaomin, and 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*, pages 389–396, New York, NY, USA, 2006. ACM.

[9] Q. Zhang and H. Mühlenbein. On the convergence of a class of estimation of distribution algorithms. *IEEE*, 8(2):127–136, April 2004.