

Evolutionary Algorithm Based on a Markov Graphical Model Selection of Promising Solutions'

Eunice Ponce de León[†], Elva Díaz[†] and Felipe Padilla[†]

[†] Electronic Systems Department, Universidad Autónoma de Aguascalientes,
Ave Universidad #940 Colonia Universidad, CP 20100, Aguascalientes,
Aguascalientes, México.
eponce@correo.uaa.mx
elva.diaz@itesm.mx
fpadilla2000@yahoo.com

Abstract. The goal of this research is to analyze how a new evolutionary algorithm based on Markov graphical model selection of promising solutions finds the optimum for some kind of functions. Probabilistic models have been used for the optimization of deceptive functions, for example, PBIL, MIMIC, BOA, UMDA, BMDA, FDA, EBNA etc. However, all mentioned algorithms are restricted in the complexity of the models used in the search. The algorithm presented in this paper, termed UEMMA (Unrestricted Evolutionary Markov Model Algorithm) was tested with the known deceptive functions. Compared with the results of the others authors the performance of this algorithm in general is not worse and in some cases is better, especially when the functions are more complex.

1 Introduction

Searching for and using a stochastic structure of the space of solutions has been an active line of research within the field of genetic and evolutionary computation. Whereas in genetic algorithms there are crossover and mutation algorithms, in Estimation of Distribution Algorithms (EDA) they have been replaced by learning and sampling of a probability distribution. This new approach for optimization emerged because the genetic algorithm paradigm needed the choice of suitable values for the parameters, and this was converted into a new optimization problem as was shown by Grefenstette [8]. This reason, together with the fact that the prediction of the movement of the population in the search space is extremely difficult, has motivated the birth of a new type of algorithm, EDA. The estimation of the joint distribution associated with the database containing the individuals of the population constitutes the bottleneck of this new heuristic [9]. The previous work began with PBIL (Population Based Incremental Learning) by Baluja [1] and later improved by Baluja and Caruana [2] where a single probability vector of binary independent variables is updated moving to the best vectors of the population, and a memory used as the information from the previous iterations is preserved. In MIMIC, a framework by De Bonnet, Isabell and Viola [4] the dependencies form a chain and the Kullback-Leibler divergence is

[†] Supported by PROMEP/UAAGS-PTC-02-01

used as measure of the fitting of the chain to the population of promising solutions. Baluja and Davies [3] uses optimal dependency trees and the Kullback-Leibler divergence is minimized. One of the most general approaches is the one proposed by Pelikan, Goldberg and Cantu-Paz, known as BOA [13]. This algorithm uses a net where each node can have k successors, allowing variables to be conditionally dependent on sets of variables. The UMDA by Mühlenbein and Pass [11] uses the same type of distribution as does PHIL algorithm, and the main difference is that a memory is used in PHIL and in UMDA the collection of vectors is used instead. The BMDA by Pelikan and Mühlenbein [14] covers second order interactions just as the case of MIMIC, however, the structure used in BMDA is more general than the one in MIMIC. Finally the FDA by Mühlenbein, Mahnig and Rodriguez [12] a factorization of the distribution is given a priori depending of the problem. The estimation of Bayesian Network Algorithm (EBNA) looks for the Bayesian Network whose structure has the maximum posterior likelihood, and whose parameter can be computed directly from the data set [7].

2 Algorithms Description

This paper presents an evolutionary stochastic algorithm that uses as a searching distribution an unrestricted structure in the class of graphic Markov Models (UMMA) as is proposed by Díaz and Ponce de León [5], defined as follows:

- choose an initial population
- determine the fitness of each individual
- perform selection of individuals to be reproduced
- repeat
 - perform crossover
 - perform mutation
 - determine the fitness of each individual
 - perform selection of individuals to be reproduced
- until some stopping criterion applies

The learning model is represented by an hypergraph and is encoded using vectors of binary variables. The graphical model selection algorithm obtains a model that best fit the population of promising solutions using an algorithm for discrete Markov model selection using the convex fitting index (CFI) based in an information measure of divergence, the Kullack-Leibler, and a penalty criteria to accomplish simplicity of

the graph [5]. This index was proposed because of its best performance compared with other two indexes, in a simulated empirical study [Díaz, Ponce de León, 2002]. The population generation uses the Gibbs-Sampler for a Graphic Markov Model defined in Díaz and Ponce de León, 2003 [6].

The UEMMA is defined as follows:

```
Generate a collection of n random vectors

Evaluate the objective function

    if the termination condition is not satisfied    continues

repeat

    order by value

    select  $\tau\%$  from the best vectors

    select and estimate the distribution (UMMA)

    generate a new population with the Gibbs-Sampler

    Evaluate the objective function

until some stopping criterion applies
```

The UEMMA generates a collection of random binary vectors in the first step, second step, evaluate the objective function, third step, if the termination condition is not satisfied continues, fourth step, order the vectors and fifth step, selects the τ percent of the best evaluated vectors. Sixth step, select the structure of the Graphic Markov Model that best accomplish the convex fitting index CFI and estimates the distribution with the UMMA algorithm and seventh step, generates a new population using the Gibbs-Sampler for a Graphic Markov Model [6], eighth step, evaluate the objective function and ninth step, if the termination condition is not satisfied go to the fourth step and repeat.

3 Function Text

3.1 OneMax Problem

This is a well-known simple linear problem used to test the convergence velocity and the scalability. It can be defined as maximizing the function:

$$F_{OneMax}(x) = \sum_{i=1}^n x_i \quad (1)$$

Where x_i is a binary variable for every i . This function has the global optimum at $(1,1,1,\dots,1)$.

3.2 Plateau Problem

This problem was proposed in Mühlenbein and Schlierkamp-Voosen [10]. The individuals of this function consists of a n -dimensional vector, such that $n=mx3$ (the genes are divided into groups of three) The function is defined as:

$$F_{Plateau}(x) = \sum_{i=1}^m g(s_i) \quad (2)$$

Where

$$g(s_i) = g(x_{3i-2}, x_{3i-1}, x_{3i}) = 1 \text{ if } x_{3i-2}=1, x_{3i-1}=1, x_{3i}=1$$

And zero in other case. This function has the global optimum at $(1,1,1,\dots,1)$.

3.3 FC₂ Problem

This function has been proposed in Muhlenbein et al. (1999) [12] and is composed by deceptive decomposable functions as follows:

$$F_{muhl}^s(x) = \begin{cases} 3.0 & \text{for } x = (0,0,0,0,1) \\ 2.0 & \text{for } x = (0,0,0,1,1) \\ 1.0 & \text{for } x = (0,0,1,1,1) \\ 3.5 & \text{for } x = (1,1,1,1,1) \\ 4.0 & \text{for } x = (0,0,0,0,0) \\ 0.0 & \text{otherwise} \end{cases} \quad (3)$$

$$F_{C_2} = \sum_{j=1}^m F_{muhl}^s(s_j) \quad (4)$$

Where $s_j = (x_{5j-4}, x_{5j-3}, x_{5j-2}, x_{5j-1}, x_{5j})$ and $n=5m$.

3.4 FC₄ Problem

This function has been proposed in Muhlenbein et al. (1999) [12] and is composed by deceptive descomposable functions as follows:

$$F^3_{cubani}(x) = \begin{cases} 0.595 & \text{for } x = (0,0,0) \\ 0.200 & \text{for } x = (0,0,1) \\ 0.595 & \text{for } x = (0,1,0) \\ 0.100 & \text{for } x = (0,1,1) \\ 1.000 & \text{for } x = (1,0,0) \\ 0.050 & \text{for } x = (1,0,1) \\ 0.090 & \text{for } x = (1,1,0) \\ 0.150 & \text{for } x = (1,1,1) \end{cases} \quad (5)$$

$$F^5_{cubani}(x) = \begin{cases} 4F^3_{cubani}(x_1, x_2, x_3) & \text{if } x_2 = x_4 \text{ and } x_3 = x_4 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$Fc_4(x) = \sum_{j=1}^m F^5_{cubani}(s_j) \quad (7)$$

Where $s_j = (x_{5j-4}, x_{5j-3}, x_{5j-2}, x_{5j-1}, x_{5j})$ and $n=5m$.

3 Experiment, results and discussion

The experiments designed consisted in fixing values of the parameters of the genetic algorithm as follows: for the percent of the best fitted bedividuals in the population t to generate the population $t + 1$ the value is 70, for the probability of selecting individuals at random for mutation the value is .1 , for the weighting coefficient in the CFI fitting index the values is .7. With the combination of factors defined, and 70 percent of the population selected for the new population in the exploration of the searching space, runs of the UEMMA for each type of function were performed as is shown in Table No. 1 .

Table 1. Experimental Results

Problem	OneMax	OneMax	Plateau	Plateau	Fc ₂	Fc ₂	Fc ₄	Fc ₄
Variables	10	10	9	9	10	10	10	10
Mean number of Generations	9	4	5	2	24	18	12	9
Time secs	60.2	47.9	85	45	193.5	116.85	84.05	82.36
Size of Population	100	300	100	300	100	300	100	300
% of Selection	70	70	70	70	70	70	70	70

The preliminary experience with this algorithm shows, as first sight that growing the size of the population diminishes the mean number of generations to obtain the optimum. The optimum was obtained in all functions. The One Max function is, of course, the easiest to optimize, and the Fc₂ is the most difficult. The time of performance is proportional to the number of generations.

4 Conclusions and recommendations

As conclusion, the performance of the algorithm UEMMA in these functions is like as reported in the literature [12]. In general the algorithm has a good performance but more experiments must be done with other functions and problems of different sizes. A comparative study with other algorithms could give new ideas, especially about the treatment of bigger problems.

As a recommendation it will be useful to design a bigger experiment calibrating together the parameters of the model selection algorithm and of the optimization algorithm in order to obtain new ideas of the relation between these two sets of parameters.

References

1. Baluja, S.: Population Based Incremental Learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, (1994).
2. Baluja, S. and Caruana, R.: Removing the genetics from the standard genetic algorithm. Technical Report, Carnegie Mellon University, (1995).
3. Baluja, S. and Davies, S.: Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In D. H. Fisher, editor, Proceedings of the 1997 International Conference on Machine Learning. Morgan Kaufman publishers (1997). Also available as Technical Report CMU-CS-97-

4. De Bonet, J. S. , Isbell, C., and Viola, P.: Mimic: Finding optima by estimating probabilities densities. *Advances in Neural Information Processing*, 9, (1996).
5. Díaz, E., Ponce de Leon, E : Discrete Graphic Markov Models by a Genetic Algorithm. In *Avances en Ciencias de la Computación e Ingeniería de Computo*. Editores: Sossa Azuela, J. H., et al. Ciudad de México (2002) 315-324.
6. Díaz, E., Ponce de León, E.: Generación aleatoria de muestras a partir de un Modelo Gráfico Markoviano. *Memorias del X Simposio de Investigación y Desarrollo Tecnológico*, Aguascalientes, (2003) 77.
7. Etxeberria, R. and Larrañaga, P. : Global optimization with Bayesian networks. In *II Symposium on Artificial Intelligence. CIMA 99. Special Session on Distributions and Evolutionary Optimization* (1999) 332-339.
8. Grefenstette, J. J. : Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1) (1998) 122-128.
9. Larrañaga, P. A Review on Estimation of Distribution Algorithms. :In *Estimation of Distributions Algorithms*. Ed. Pedro Larrañaga and José Lozano. Kluwer Academic Publishers, (2002).
10. Mühlenbein, H. and Schierkamp-Voosen, D. : The science of breeding and its application to the breeder genetic algorithm (BGA). *Evolutionary Computation*, 1 (1993) 335-360.
11. Mühlenbein, H. and Pass, G.: From recombination of genes to the estimation of distributions 1. binary parameters. In A. E. Eiben, T. Bäck, M. Shoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature—PPSN V*, Springer, (1998) 178-189.
12. Mühlenbein, H., Mahnig, T. and Ochoa-Rodriguez, A.: Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5, (1999) 215-247.
13. Pelikan, M., Goldberg, D. E. and Cantu-Paz E.: Boa The Bayesian optimization algorithm. In W. Banzhaf, et al., editors, *Proceedings of the GECCO-1999 Genetic Evolutionary Computation Conference*. Morgan Kaufman Publishers, (1999) 525-532.
14. Pelikan, M. and Mühlenbein, H.: The bivariate marginal distribution algorithm. *Advances in Soft Computing-Engineering Design and Manufacturing* (1999) 521-535.