Analysis of the Univariate Marginal Distribution Algorithm modeled by Markov chains

C. González, J. D. Rodríguez, J. A. Lozano, and P. Larrañaga

Department of Computer Science and Artificial Intelligence,
University of the Basque Country, Spain
{ccpgomoc,scsrofej,ccploalj,ccplamup}@sc.ehu.es

Abstract. This work presents an analysis of the convergence behaviour of the Univariate Marginal Distribution Algorithm (UMDA) when it is used to maximize a number of pseudo-boolean functions.
The analysis is based on modeling the algorithm using a reducible Markov chain, whose absorbing states correspond to the individuals of the search space. The absorption probability to the optimum and the expected time of convergence to the set of absorbing states are calculated for each function. This information is used to provide some insights into how the absorption probability to the optimum and the expected absorption times evolve when the size of population increases. The results show the different behaviours of the algorithm in the analyzed functions.

1 Introduction

Estimation of Distribution Algorithms (EDAs) constitute a new and promising paradigm for EAs [5, 9]. Introduced by Mühlenbein and Paaβ [9], EDAs are based on Genetic Algorithms (GAs) and constitute an example of stochastic heuristics based on populations of individuals, each of which encodes a possible solution of the optimization problem. These populations evolve in successive generations as the search progresses, organized in the same way as most Evolutionary Computation heuristics. In contrast to GAs, which consider the crossover and mutation operators as essential tools to generate new populations, EDAs replace those operators by estimating and sampling the joint probability distribution of the selected individuals.

Unfortunately, the bottleneck of this new heuristic lies in estimating the joint probability distribution associated with the database containing the selected individuals. To avoid this problem, several authors have proposed different algorithms where simplified assumptions concerning the conditional dependencies between the variables of the joint probability distribution are made. A review of different approaches in the combinatorial and numerical fields can be found in [4, 5].

The purpose of this this paper is to further investigate the convergence behaviour of the simplest EDA –UMDA–, which is applied to the maximization of a number of pseudo-boolean functions.

J. Mira and J.R. Álvarez (Eds.): IWANN 2003, LNCS 2686, pp. 510-517, 2003.

The analysis is based on modeling the algorithm using a Markov chain whose absorbing states correspond to the individuals of the search space. Hence some natural questions immediately arise. What is the absorption probability to any absorbing state (particularly to the optimal point)? How long must we wait until the set of absorbing states are visited? Answering these questions will enable us to learn the effects that changes in the individual and population size have on the absorption probability to the optimum and the expected absorption times. In order to do so we calculate those quantities for UMDA on the maximization of an example of linear, pseudo-modular, unimax and almost positive functions, for different values of population size N and individual length l. Due to the high computational cost of these calculations we have to use low values of N and l.

The remainder of this paper is organized as follows: Section 2 introduces the UMDA algorithm. In Section 3 the Markov chain that models the algorithm is described, and some useful theoretical results are revised. The studied functions are introduced in Section 4. Section 5 explains the experiments carried out and analyzes the results. Finally, we draw conclusions in Section 6.

2 The UMDA algorithm

The Univariate Marginal Distribution Algorithm was proposed by Mühlenbein [7] in 1998. A pseudocode for this algorithm can be seen in Figure 1. UMDA

UMDA
  $D_0 \leftarrow$ Generate M individuals (the initial population) randomly
  Repeat for $t = 1, 2, \ldots$ until the stopping criterion is met
    $D_{t-1}^{Se} \leftarrow$ Select the $N \leq M$ individuals from D_{t-1} according to the selection method
    $p_t(\mathbf{x}) = p(\mathbf{x}|D_{t-1}^{Se}) = \prod_{i=1}^{l} \frac{\sum_{j=1}^{N} \delta_j(X_i = x_i|D_{t-1}^{Se})}{N} \leftarrow$ Estimate the joint probability distribution
    $D_t \leftarrow$ Sample M individuals (the new population) from $p_t(\mathbf{x})$

Fig. 1. Pseudocode for a general UMDA algorithm.

uses the simplest model to estimate the joint probability distribution of the selected individuals in each generation, $p_t(\mathbf{x})$. This joint probability distribution is factorized as a product of independent univariate marginal distributions, $p_t(\mathbf{x}) = p(\mathbf{x}|D_{t-1}^{Se}) = \prod_{i=1}^{l} p_t(x_i)$.

Each univariate marginal distribution is estimated from marginal frequencies, $p_t(x_i) = \frac{\sum_{j=1}^{N} \delta_j(X_i = x_i|D_{t-1}^{Se})}{N}$, where:

$$\delta_j(X_i = x_i|D_{t-1}^{Se}) = \begin{cases} 1 & \text{if in the } j\text{-th case of } D_{t-1}^{Se},\ X_i = x_i \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

In practice UMDA is used with elitism and the estimation of the parameters is carried out by means of Laplace correction [1]: $p_t(x_i) = p(x_i|D_{t-1}^{Se}) = \frac{\sum_{j=1}^{l} \delta_j(X_i=x_i|D_{t-1}^{Se})+1}{N+2}$. This ensures convergence to the optimum [2].

Previous works on UMDA [6, 8] have been carried out by assigning a dynamical system to the algorithm, and showing that the algorithm can converge to any local optima of the search space. However this result is subject to the use of an infinite population, far from the finite population case, where the algorithm can converge to any point of the search space.

We analyze UMDA with finite population in order to provide new insights into the circumstances under which UMDA will (will not) perform well.

3 Using Markov chains to model UMDA

Let us introduce some notation. The search space is represented by $\Omega = \{0,1\}^l$, where $l \in I\!N$. The cardinality of the search space is $|\Omega| = 2^l = n$. We consider the optimization problem $\max_{\mathbf{x}\in\Omega} f(\mathbf{x})$, where $f : \Omega \longrightarrow I\!R$ is the objective function.

We use a specific version of UMDA to solve the above problem. The algorithm works as follows: at each step t we have a population of size $M = 2N$, D_{t-1}, from which we select the N best individuals (truncation selection), obtaining D_{t-1}^{Se}. Later, using these selected individuals the joint probability distribution $p_t(\mathbf{x})$ is estimated as in Figure 1. Finally we obtain the new population D_t sampling $2N$ individuals from $p_t(\mathbf{x})$.

Given that the probability distribution at step t only depends on the probability distribution at step $t-1$, the UMDA algorithm above can be modeled using a Markov chain, where the states of the chain are the different probability distributions the algorithm can take. If we take into account that each probability distribution can be represented as a probability vector $\mathbf{q} = (q_1, \ldots, q_l)$ (where q_i is the probability of obtaining a 1 in the ith gene), the set of states can be expressed as follows:

$$E = \left\{ (q_1, \ldots, q_l) \quad | \quad q_i \in \{0, \frac{1}{N}, \ldots, \frac{N-1}{N}, 1\}, \quad i \in \{1, \ldots, l\} \right\}. \qquad (2)$$

The cardinality of the state space is $c = |E| = (N+1)^l$. It is important to note that the cardinality of the space states c increases exponentially as the individual size does. This is the reason why, in order to have a reasonable computational cost, the experiments will be carried out for small values of l.

We also want to stress that the Markov chain is not irreducible. More precisely, the absorbing states of the Markov chain correspond to the individuals of the search space while the transient states are the rest, i.e. the states with some component not equal to 0 or 1.

To aid in comprehension, note that each absorbing state is associated with a uniform population of selected individuals (which is formed by N copies of

the same individual). Taking into account that the absorbing states are the individuals of the search space, the algorithm could converge to any of them.

Calculation of the absorption probabilities and the expected absorption times

Let's suppose that the states of the Markov chain are ordered in such a way that the absorbing states are in the last places. Therefore the transition probability matrix P associated with the Markov chain can be written as follows:

$$P = \begin{pmatrix} Q & R \\ \emptyset & I \end{pmatrix} \tag{3}$$

where $\emptyset$ is the null matrix, and I is the identity matrix.

The formulas for the absorption probability and expected absorption times can be obtained from matrices: $W = (I - Q)^{-1}$ and $U = WR$. These results can be seen in [10].

The *expected absorption time starting from the ith state* v_i is given by the expression $v_i = \sum_j w_{ij}$.

The *absorption probabilities to an absorbing state* j *starting from the ith state*, u_{ij} are given by the elements of the matrix $U = (u_{ij})$.

The computational cost of the calculation of the above quantities depends on the cost of inverting W. Since the dimension of W is $(N+1)^l - 2^l$, it is directly related to the individual size l.

Calculation of the transition probability matrix P

The calculation of the transition probability matrix P is basic to obtain the remaining quantities. Each entry of $P = (p_{ij})$ is the probability of going from state $\mathbf{q}_i$ to state $\mathbf{q}_j$ in a step of the algorithm, and can be obtained as follows:

$$p_{ij} = P(\mathbf{q}_j|\mathbf{q}_i) = \sum_D P(\text{obtain population } D| \text{ the vector } \mathbf{q}_i \text{ is sampled}) \ , \tag{4}$$

where D varies in the populations that can be obtained from $\mathbf{q}_i$, and from the selected individuals D^{Se} of D, the probability vector $\mathbf{q}_j$ is obtained. Unfortunately, in order to calculate (4) it is necessary to solve a system of equations with a large number of degrees of freedom (which depends on l), which has a high computational cost. This is the reason why we use a different method to estimate the elements of matrix P. In Section 5.1 we explain in detail how the estimation was carried out.

4 The functions used

The particular **linear function** analyzed in this work is $f(\mathbf{x}) = c_0 + \sum_{i=1}^{l} c_i x_i$, $x_i \in \{0,1\}$, $c_i \in I\!R$ such that $c_i > \sum_{j=0}^{i-1} c_j$. It is clear that $(1,\ldots,1)$ is the only global maximum for this function.

The **pseudo-modular function** we have used is $f(\mathbf{x}) = \sum_{i=1}^{l} \prod_{j=1}^{i} x_j$, $x_j \in \{0,1\}$, whose optimal solution is $(1,\ldots,1)$.

In the experiments we have carried out we used a well known **unimax function**, the long path function [3] (we can not include it here for reasons of space). We want to stress that this function only has sense for odd values of l.

The **almost positive** function analyzed in this paper is $f(\mathbf{x}) = l - \sum_{i=1}^{l} x_i + (l+1) \prod_{i=1}^{l} x_i, x_i \in \{0,1\}$. The optimal solution is $(1,\ldots,1)$, while the individual $(0,\ldots,0)$ is a local optimum point.

5 Experimental results

Our aim is to find the absorption probability to the optimum and the expected absorption times to some absorbing states when the UMDA algorithm is used to maximize the pseudo-boolean functions introduced in the previous section. Once we have those quantities we analyze how they evolve when the size of population N varies. We have made our analysis when $l = 2$ and $2 \leq N \leq 8$, and when $l = 3$ and $3 \leq N \leq 8$.

Estimation of the transition probability matrix P

The computational complexity of the exact calculation of $P = (p_{ij})$ forces us to estimate these values instead of carrying out an exact calculation. To obtain the values of the jth row of P corresponding to probability vector $\mathbf{q}_j$, we carry out k times the following two steps for each j:

- $\mathbf{q}_j$ is taken as the initial vector of probabilities, each time carrying out the basic steps of UMDA: (i) The initial population is obtained sampling $\mathbf{q}_j$, (ii) The N best individuals are selected, giving us the selected population, (iii) The new vector of probabilities $\mathbf{q}_i$ is obtained from the selected population.
- After the previous step the obtained probability vector is picked up.

If we denote by k_i the number of times that we reach the state $\mathbf{q}_i$, then each value p_{ij}, $i \in \{1,\ldots,c\}$ of row j is estimated as $p_{ij} = \frac{k_i}{k}$.

It is clear that when the number of experiments carried out k increases the estimation improves. In our experiments we chose k in order to obtain a reasonable computational cost. We made a number of $k = 5,000,000$ experiments which fixes the 5th decimal of p_{ij}.

The absorption probability to the optimum and the expected absorption time to some absorbing state

In practice, the first probability vector used to be $(1/2,\ldots,1/2)$, so we have calculated the absorption probability (resp. time) to the optimum from this state when N is even. In case of odd N we have used the mean of the probability vectors that are the nearest to $(1/2,\ldots,1/2)$ (see Table 1).

Summarizing the results

The results can be seen in Figure 3. The absorption probability to the optimum (and the expected absorption time to some absorbing state) is given in a graph, where the y axis shows the absorption probability (resp. the expected absorption time) and the x axis shows the size of the population N. The same graph shows the results for $l = 2$ and $l = 3$.

l/ N	2	3
2	$(\frac{1}{2},\frac{1}{2})$	
3	$(\frac{1}{3},\frac{2}{3})$ $(\frac{2}{3},\frac{1}{3})$	$(\frac{1}{3},\frac{1}{3},\frac{2}{3})$ $(\frac{1}{3},\frac{2}{3},\frac{1}{3})$ $(\frac{2}{3},\frac{1}{3},\frac{1}{3})$ $(\frac{1}{3},\frac{2}{3},\frac{2}{3})$ $(\frac{2}{3},\frac{1}{3},\frac{2}{3})$ $(\frac{2}{3},\frac{2}{3},\frac{1}{3})$
4	$(\frac{2}{4},\frac{2}{4})$	$(\frac{2}{4},\frac{2}{4},\frac{2}{4})$
5	$(\frac{2}{5},\frac{3}{5})$ $(\frac{3}{5},\frac{2}{5})$	$(\frac{2}{5},\frac{2}{5},\frac{3}{5})$ $(\frac{2}{5},\frac{3}{5},\frac{2}{5})$ $(\frac{3}{5},\frac{2}{5},\frac{2}{5})$ $(\frac{2}{5},\frac{3}{5},\frac{3}{5})$ $(\frac{3}{5},\frac{2}{5},\frac{3}{5})$ $(\frac{3}{5},\frac{3}{5},\frac{2}{5})$
6	$(\frac{3}{6},\frac{3}{6})$	$(\frac{3}{6},\frac{3}{6},\frac{3}{6})$
7	$(\frac{3}{7},\frac{4}{7})$ $(\frac{4}{7},\frac{3}{7})$	$(\frac{3}{7},\frac{3}{7},\frac{4}{7})$ $(\frac{3}{7},\frac{4}{7},\frac{3}{7})$ $(\frac{4}{7},\frac{3}{7},\frac{3}{7})$ $(\frac{3}{7},\frac{4}{7},\frac{4}{7})$ $(\frac{4}{7},\frac{3}{7},\frac{4}{7})$ $(\frac{4}{7},\frac{4}{7},\frac{3}{7})$
8	$(\frac{4}{8},\frac{4}{8})$	$(\frac{4}{8},\frac{4}{8},\frac{4}{8})$

Table 1. Starting probability vectors chosen.

Several comments can be made in view of the graphs. We can distinguish the behaviour of the first three functions (the easiest to optimize) from the almost positive.

As was expected, for the first three functions, the absorption probability increases with N. This probability is near to one in the linear and pseudo-modular functions, which means that in almost all executions the algorithm will converge to the optimum. On the other hand, in the unimax function this probability is lower than 0.6. However, it seems that the growth of this probability with N is higher in this third function. Similarly this probability is smaller when l is bigger. The small number of generations to reach convergence is noteworthy.

We obtained surprising results regarding the expected absorption times. In the linear and pseudo-modular functions this time does not increase with population size. We think it is related to the absorption probability. Because the absorption probability is higher the algorithm converges faster. On the other hand in the unimax function this time increases linearly with N. In this last case the function is harder to optimize than the others so the algorithm needs more time to converge. The same as before, absorption time increases with l.

The case of the almost positive function is the most interesting. While with $l = 2$ the absorption probability increases a little with N, in dimension 3 this probability decreases. Apparently going to 0 with N. It seems the algorithm can be absorbed by the local optimum point $(0, \ldots, 0)$. So this function is hardly optimized with UMDA.

6 Conclusions and future work

In this work we have used Markov chains to model and analyze some interesting questions about UMDA algorithm behaviour on pseudo-boolean functions. We hope that further theoretical studies on the behaviour of UMDA can be based on this work.

For each analyzed function, we have calculated the absorption probabilities to the optimal point and the expected absorption times. This calculation enables us to see the effects that changes in population size have on these two quantities. The analysis shows the behaviour of the algorithm when the complexity of the function increases: the absorption probability decreases while the expected

absorption time increases. Even in the almost positive function the absorption probability goes near to zero when N increases.

There is much further work to be done to increase our understanding of the EDAs algorithm's behaviour on different classes of problems. A first task would be to increase both individual length and population size. Another interesting direction would be to explore the relationship between the probability of reaching the optimum or any other point of the search space. It would also be helpful to arrive at an analogous model and analysis for other EDAs.

7 Acknowledgments

This work was supported by the University of the Basque Country under grant no. 9/UPV/EHU 00140.226-12084/2000. Also C. González is supported by UPV-EHU.

References

1. B. Cestnik. Estimating Probabilities: A Crucial Task in Machine Learning. In *Proceedings of the European Conference in Artificial Intelligence*, pages 147–149, 1990.
2. C. González, J. A. Lozano, and P. Larrañaga. Mathematical Modelling of Discrete Estimation of Distribution Algorithms. In P. Larrañaga and J. A. Lozano, editors, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, pages 147–163. Kluwer Academic Publishers, 2002.
3. J. Horn, D. E. Goldberg, and K. Deb. Long Path Problems. In Y. Davidor, H. P. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature, PPSN*, volume 3, pages 149–158. Berlin and Heidelberg: Springer, 1994.
4. P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña. Combinatorial Optimization by Learning and Simulation of Bayesian Networks. In C. Boutilier and M. Goldszmidt, editors, *Proceedings of Uncertainty in Artificial Intelligence, UAI-2000*, pages 343–352. Morgan Kaufmann, 2000.
5. P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation.* Kluwer Academic Publishers, 2002.
6. T. Mahnig and H. Mühlenbein. Mathematical Analysis of Optimization Methods Using Search Distributions. In A. S. Wu, editor, *Proceedings of the 2000 Genetic and Evolutionary Computation Conference, Workshop Program*, pages 205–208, 2000.
7. H. Mühlenbein. The Equation for Response to Selection and its Use for Prediction. *Evolutionary Computation*, 5:303–346, 1998.
8. H. Mühlenbein and T. Mahnig. Evolutionary Computation and Wright's Equation. *Theoretical Computer Science (in press)*, 2001.
9. H. Mühlenbein and G. Paaβ. From Recombination of Genes to the Estimation of Distributions I. Binary Parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature, PPSN-IV*, pages 178–187, 1996.
10. H. M. Taylor and S. Karlin. *An Introduction to Stochastic Modeling.* Academic Press, 1993.

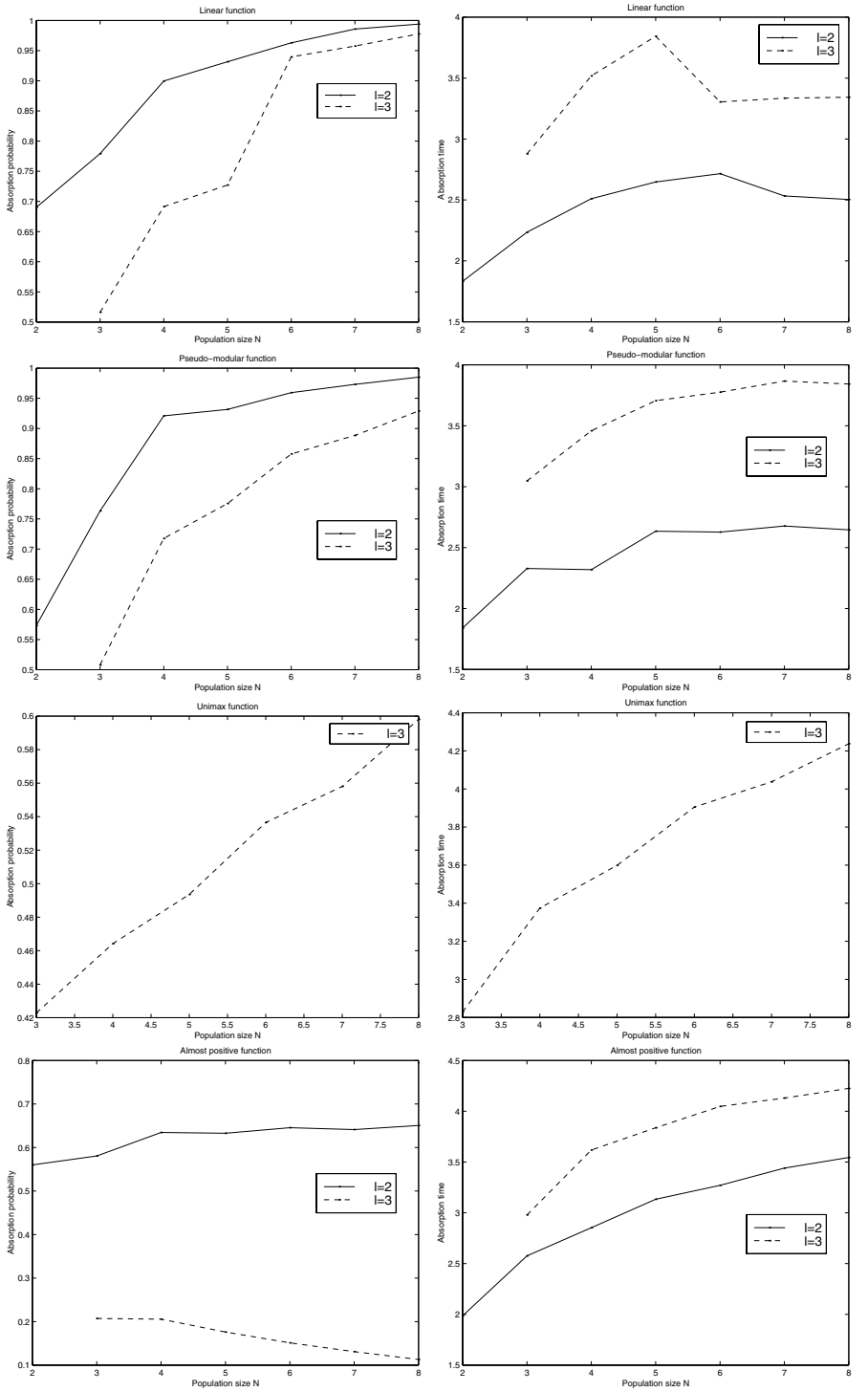

Fig. 2. Absorption probability and absorption time for the linear, pseudo-modular, unimax and almost positive functions.