Regularized k-order Markov models in EDAs

Roberto Santana
Computational Intelligence Group
Universidad Politécnica de Madrid, Spain
roberto.santana@upm.es

Hossein Karshenas
Computational Intelligence Group
Universidad Politécnica de Madrid, Spain
hkarshenas@fi.upm.es

Concha Bielza
Computational Intelligence Group
DIA, Universidad Politécnica de Madrid, Spain
mcbielza@fi.upm.es

Pedro Larrañaga
Computational Intelligence Group
DIA, Universidad Politécnica de Madrid, Spain
pedro.larranaga@fi.upm.es

ABSTRACT

k-order Markov models have been introduced to estimation of distribution algorithms (EDAs) to solve a particular class of optimization problems in which each variable depends on its previous k variables in a given, fixed order. In this paper we investigate the use of regularization as a way to approximate k-order Markov models when k is increased. The introduced regularized models are used to balance the complexity and accuracy of the k-order Markov models. We investigate the behavior of the EDAs in several instances of the hydrophobic-polar (HP) protein problem, a simplified protein folding model. Our preliminary results show that EDAs that use regularized approximations of the k-order Markov models offer a good compromise between complexity and efficiency, and could be an appropriate choice when the number of variables is increased.

Categories and Subject Descriptors

G.1 [**Optimization**]: Global optimization; G.3 [**Probabilistic methods**]

General Terms

Algorithms

Keywords

Estimation of distribution algorithms, probabilistic modeling, HP protein model, Markov models

1. INTRODUCTION

Evolutionary algorithms (EAs) usually implicitly exploit the relationships between the variables to discover promising areas of the search space. Traditional EAs such as genetic algorithms (GAs) employ genetic operators for this purpose. Advanced EAs, such as estimation of distribution algorithms (EDAs) [13, 15, 16] learn a probabilistic model that explicitly models these relationships. EDAs have been successfully applied to a variety of problems and they have been shown to be particularly suitable for problems where strong dependencies between the variables arise.

One particular class of EDAs uses models where each variable depends on the k, $k \in \mathcal{N}$, previous variables in a given order. These models, that can be seen as a generalization of chain shaped distributions [6], for which $k = 1$, were called in [18] k-order Markov models. The term refers to the analogy with Markov chains in which the state of variable X_i depends on the states of its previous k variables in the chain. This type of Markov models, which are different to Markov networks also applied in EDAs [17, 22], have been reported to be appropriate for problems where the assignment to a given variable can be made to mainly depend on the assignments of the previous k variables [2, 3, 18, 20]. Since EDAs that use these models do not require structural learning, the EDA learning step can be very efficient, particularly when k is small ($k \leq 3$). However, as k is increased, the computational complexity of the algorithm grows exponentially. This is so because the size of the conditional probability tables associated to each random variable is exponential in k. Therefore, in practice it is not feasible to investigate the suitability of Markov models for higher values of k.

In this paper, we propose an alternative approach for modeling the type of relationships the k-order Markov models are intended for. Our approach is based on the idea of using regression to predict each variable given some combination of its previous k variables. Therefore, instead of learning a conditional probability table for each variable, the parameters of the regression are computed. In addition, regression is applied using regularization methods [8, 9, 14, 25]. In regularization, the model estimation process is "regularized" by using a specific penalization term on the values of these parameters. Although regularization has been applied in many ways, and in several different contexts, in most of the cases it is applied to regression formulas of the model parameters, its score or its probability parameters (if they are different).

GECCO'11, July 12–16, 2011, Dublin, Ireland.

In our approach to learn regularized models, the elastic-net [25], a regularization and variable selection method which encourages a grouping effect where strongly correlated predictors tend to be in or out of the model together, is used in the framework of multinomial regression. Different variants of regularized models have been recently proposed in EDAs for discrete [23] and continuous problems [11].

To evaluate the performance of the introduced algorithms we use several instances of the hydrophobic-polar (HP) model [7]. This model is based on the fact that hydrophobic interactions are a dominant force in protein folding. The HP model has arisen as a suitable benchmark for cross-disciplinary studies involving domains such as computational biology, statistical and chemical physics and optimization. In the optimization domain, the search for the protein structure is transformed into the search for the optimal configuration given an energy function that takes into account the HP interactions that arise in the model. The problem of finding such a minimum energy configuration is NP-complete for the 2-d [5] and 3-d [1] lattices.

The paper is organized as follows. In the next section, k-order Markov probability models are reviewed. In Section 3, the regularized approximation of the k-order Markov probability models is introduced. Section 4 explains the main components of the EDAs based on regularized Markov models. The HP protein model is introduced in Section 5. The experimental framework and the numerical results are presented in Section 6. The conclusions and lines for future research are discussed in Section 7.

2. K-ORDER MARKOV MODELS

We use X_i to represent a random variable. A possible value of X_i is denoted x_i. Similarly, we use $\mathbf{X} = (X_1, \ldots, X_n)$ to represent an n-dimensional random variable and $\mathbf{x} = (x_1, \ldots, x_n)$ to represent one of its possible values.

Given an ordering of the variables, in the k-order Markov model [18] the configuration of variable X_i depends on the configuration of all the previous k variables, where $k \geq 0$ is a parameter of the model. When $k > 0$, the joint probability distribution can be factorized as follows:

$$p_{MK}(\mathbf{x}) = p(x_1, \ldots, x_{k+1}) \prod_{i=k+2}^{n} p(x_i \mid x_{i-1}, \ldots, x_{i-k}) \quad (1)$$

otherwise, $p_{MK}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i)$. The complexity of the model depends on parameter k. Figure 1 shows k-order Markov models of different complexity. Notice, that the independence model can be representing by setting $k = 0$.

The pseudocode of the Markov EDA (Mk-EDA$_k$) is shown in Algorithm 1. The main step is the parametric learning of the probabilistic model. Since the structure of the Markov model is given, this step comprises to calculate the frequencies from the set of selected individuals and to compute the marginal and conditional probabilities. To sample a solution, first variables in the factor $(x_1, \ldots, x_{k+1})$ are generated and the rest of variables are sampled according to the order specified by the Markov factorization.

3. REGULARIZED MARKOV MODELS

Our aim is to find feasible approximations of Markov models where the number of parameters required to approximate the dependence between each variable X_i and previous k variables could be diminished. The approach will be regressing X_i in terms of a combination of its previous k values. To further enforce sparsity in the number of parameters regularization is applied. In the following sections, we first explain the regularization method of choice and then the way the parameters that encode the dependence between X_i and its k related variables are defined.

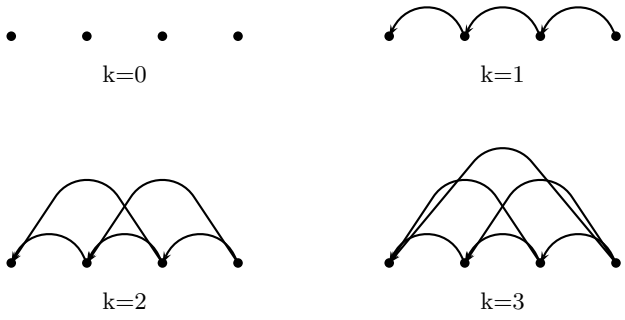

Figure 1: k-order Markov models of different complexity.

Algorithm 1: **Markov-EDA**

1 $D_0 \leftarrow$ Generate M individuals randomly and evaluate them
2 $t = 1$
3 **do** {
4 $\quad D^s_{t-1} \leftarrow$ Select $N \leq M$ individuals from D_{t-1} according to a selection method
5 $\quad$ Compute the marginal and conditional probabilities corresponding to each factor of factorization (1)
6 $\quad D_t \leftarrow$ Sample M individuals (the new population) from the k-order Markov model
7 } **until** A stop criterion is met

3.1 Regularized multi-logit regression

We consider the general case where the response variable can have m possible values, i.e., the cardinality of X_i is $m \geq 2$. In this case, the multi-logit model is expressed as:

$$log \frac{Pr(X_i = l|\mathbf{y})}{Pr(X_i = m|\mathbf{y})} = \beta_{0l} + \mathbf{y}^T \beta_\mathbf{l}, l = 1, \ldots, m-1 \quad (2)$$

where β_{l0} and β_l are the parameters of the linear model for class l, and $\mathbf{y}$ is a p-vector of predictor variables. The way in which predictor variables are selected is key to our proposal and it is explained in the next section.

Following [8, 24],

$$Pr(X_i = l|\mathbf{y}) = \frac{e^{\beta_{0l} + \mathbf{y}^T \beta_l}}{\sum_{j=1}^{m} e^{\beta_{0j} + \mathbf{y}^T \beta_j}} \quad (3)$$

The model is fitted using the regularized maximum multi-logit likelihood by means of the elastic net approach [25]. This is an algorithm applied in different domains, that allows to combine the lasso and ridge regularization and for which an efficient implementation was available.

Let N be the number of observations, and $\mathbf{x}^j, j \in \{1,\ldots,N\}$, be the jth vector of variables. In addition, let $p_l = Pr(X_i = l, \mathbf{y}^j)$ and $g_j = x_i^j \in \{1,\ldots,m\}$. The penalized loglikelihood

$$\left[\frac{1}{N}\sum_{j=1}^{N} logp_{g_j}(\mathbf{y}^j) - \lambda \sum_{l=1}^{m} P_\alpha(\beta_l)\right], \quad (4)$$

where

$$P_\alpha(\beta) = (1-\alpha)\frac{1}{2}||\beta||_{l_2}^2 + \alpha||\beta||_{l_1} \quad (5)$$

$$= \sum_{j=1}^{p}\left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta|_j\right] \quad (6)$$

is maximized over $\{\beta_0, \beta\}_1^m \in \mathbb{R}^{m(p+1)}$ using the elastic net [25].

$P_\alpha(\beta)$ is a compromise between the ridge-regression penalty ($\alpha = 0$) and the lasso penalty ($\alpha = 1$). We have used the implementation proposed in [8] which computes the models using cyclical coordinate descent, applied along the regularization path. More details about the model and the implementation can be found in [25].

3.2 Selection of the predictor variables

We consider three different variants to select the predictor variables. Each variant corresponds to a model of different complexity.

- Rgk: This is the simplest case. We will predict the value of variable X_i given its previous k variables in the ordering. Therefore, in this simple scenario $\mathbf{y} = (x_{i-1},\ldots,x_{i-k})$. The idea of this approximation is to estimate the assignment of X_i as a linear combination of each previous k variables.

- BivRgk: We also consider a more complex approximation in which X_i is expressed as a combination of all pairwise interactions between its previous k variables. In this case, $\mathbf{y} = (\bigcup_{j=3}^{j=k} x_{i-2}x_{i-j},\ldots,x_{i-k+1}x_{i-k})$ and $\bigcup$ is interpreted as the concatenation operator of all the pairs in a vector.

- AllRgk: This corresponds to the most complex model. X_i can be expressed as a combination of its previous k variables, and their corresponding pairwise interactions, i.e.:

$$\mathbf{y} = (\bigcup_{j=2}^{j=k} x_{i-1}x_{i-j}, \bigcup_{j=3}^{j=k} x_{i-2}x_{i-j},\ldots,x_{i-k+1}x_{i-k}) \quad (7)$$

The order of the maximum number of parameters needed to estimate the models of Rgk, BivRgk and AllRgk are, respectively: $O(k)$, $O(k^2)$, and $O(k^2)$ which compares favorably with $O(m^k)$ which is the number of parameters that would be needed to represent the joint probability table. Also notice, that the regularization will set to zero those parameters that do not contribute to the prediction. Therefore, the number of required parameters can be further reduced.

By only including the estimation of pairwise interactions, the prediction may be less accurate than if higher order interactions were included. However, as shown in our experiments, in some cases the approximation may be sufficient to work in the context of the EDA optimization approach.

4. REGULARIZED MARKOV MODELS IN EDAS

The pseudocode of the EDA with regularized models (MkRg-EDA) is shown in Algorithm 2. The algorithm starts by randomly sampling a population of points which are evaluated according to the fitness function. Selection is accomplished based on the fitness function of the individuals. In this paper we use truncation selection although other selection methods can be used. The fundamental steps of MkRg-EDA are the learning method (step 5), which receives as input the selected population and outputs a set of local probabilistic models and the sampling method (step 6), which receives the set of probabilistic models and its parameters and outputs a new generated population. These two steps are respectively described in Algorithms 3 and 4. We use the maximum number of generations as stop criterion.

Algorithm 2: **MkRg-EDA**

1	$D_0 \leftarrow$ Sample M individuals from a random uniform distribution and evaluate them
2	t $\leftarrow 1$
3	**do** {
4	$D_{t-1}^{Se} \leftarrow$ Select N individuals from D_{t-1}
5	For each variable X_i, learn a regularized model of X_i given its predictors
6	Using the set of regularized models, sample M new individuals and evaluate them
7	} **until** Stop criterion is met

4.1 Learning

The elastic net procedure (Algorithm 3) computes the predictions of a given target variable for a decreasing set of lambda values (lambda sequence). This means that for each variable, the actual output of the method is a parameterized set of predictions, one set of regression coefficients for each lambda. We choose the lambda value that minimizes the square error between the variable values and the predictions.

Algorithm 3: **MkRg-EDA learning**

1	For $i = 1$ to n
2	Compute the predictors of X_i from the set of previous variables $X_{i-1},\ldots,X_{i-t}$, $t = max(1, i-k)$
3	Using the elastic-net procedure, learn a regularized model of X_i given its predictors

We propose three different variants of MkRg-EDA that will respectively use the Rgk, BivRgk and AllRgk models described in Section 3.2. In the experiments, we will identify the variants of MkRg-EDA as MkRgk, MkBivRgk, and MkAllRgk. Two different k parameters are used for each regularized EDA, $k = 3$ and 5. In the experiments, the six regularized k-Markov EDAs are compared to Mk-EDA$_1$, Mk-EDA$_2$ and Mk-EDA$_3$.

4.2 Sampling

The implementation of the sampling step is relatively simple. The regularized models assign a value to each variable given their predictor values. However, the variables that are

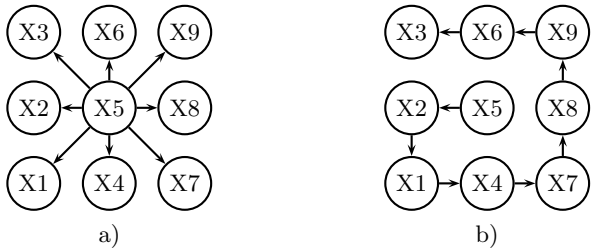

Figure 2: Two different types of probabilistic graphical models according to their topology. (a) Central model. (b) Markov model $k = 1$.

first in the sampling order will require some initial assignment to be given to their predictors. This is a particular case of the "sampling ordering" issue, that arises in sampling methods where a sampling order that guarantees that all the dependencies will be used for sampling is difficult or impossible to find [17, 22].

To sample the initial (k) variables we propose the following general alternatives: 1) Random initialization. 2) Use a set of solutions previously evaluated, selected or full population at time $t-1$. 3) Initialization from sampling simpler probabilistic models. 4) Random initialization followed by the application of a local optimizer.

Randomly initialized values guarantees no bias in the sampling process, but the values maybe too far from the optimal ones, delaying the convergence of the algorithm. Using the previous population will improve the quality of the initial solutions but the algorithm may be biased towards regions of the space already explored. Initialization from a simpler probabilistic (e.g. a univariate probabilistic model) also learned from the selected population, outperforms random initialization in terms of the solution quality but learning of a second model cannot be affordable in terms of time. Finally, we can generate the values corresponding to the initial variables randomly and then apply some local optimizer on these values. For the experiments presented in this paper, we used the second method, initializing from the previous selected population which is used as selection pool to choose initial solutions for sampling.

Algorithm 4: **ENReg-EDA sampling**

1	Create the initial population randomly picking solutions from the previous selected population
2	For $j = 1$ to n
3	Use the regularized model of X_i to estimate its value x_i from its predictors
4	Set $X_i = x_i$
5	Apply the repair operator.

After solutions have been sampled, the repair operator proposed in [3] is applied. The idea of the repair operator is to correct self-intersecting solutions produced by sampling. It is a problem specific operator. The method introduced in [3] is more efficient that the one originally proposed in [4] and subsequently applied in [18, 20, 19]. This backtracking algorithm guarantees that the HP sequence will not be selfintersecting.

Finally, we discuss a characteristic of Markov models, related with the sampling process, and that makes a difference to other classes of probabilistic models used in EDAs. We call this issue, the *propagation error question*. It refers to the potential unequal distribution of sampling errors due to the propagation of early errors during sampling. To illustrate this issue, Figure 2 shows two different models according to their topology. In the "central model" shown in Figure 2a), all the variables except X_5 have the same probability of having a wrong assignment during sampling.

We assume that the sampling error of a variable is the expected difference between a probability sample value and the true value. In the central model, the sampling of all variables except X_5 are independent of each other and uniquely depends on X_5. On the contrary, in the Markov model shown in Figure 2b), the assignment of variable X_2 depends on X_5 but every other variable depend on the assignments of the previous variables in the order (notice that in this case the order of the model does not agree with the enumeration of variables). As a consequence, it is more likely to have a sampling error for variable X_3 than for variable X_2. The influence of the *propagation error question* in the behavior of the EDAs that use these Markov models is an open question.

5. HP PROTEIN MODEL

Under specific conditions, a protein sequence folds into a native 3-d structure. The problem of determining the protein native structure from its sequence is known as the protein structure prediction problem. To solve this problem, a protein model is chosen and an energy is associated to each possible protein fold. The search for the protein structure is transformed into the search for the optimal protein configuration given the energy function.

The HP simplified protein model [7] is used in bioinformatics to investigate protein folding. In the HP model, a protein is considered a sequence of hydrophobic (H) and hydrophilic or polar (P) residues which are located in regular lattice models forming self-avoided paths. In the optimization of the HP-protein model 2- and 3-dimensional lattices are the most commonly used. Figure 3 shows the graphical representations of two possible configurations for sequence $HHHPHPPPPPH$ in 2 dimensions.

Interactions between neighbor residues (adjacent in the lattice but not connected in the sequence) contribute to the total energy of the HP lattice configuration. The energy values associated with the functional HP model [10] contain both attractive $\epsilon_{HH} = -2$ and repulsive interactions ($\epsilon_{PP} = 1$, $\epsilon_{HP} = 1$, and $\epsilon_{PH} = 1$). The HP problem consists of finding the solution (HP chain topological configuration) that minimizes the total energy. The energy that the functional model protein associates with the configuration shown in Figure 3a) is 1 because there is one HH interaction, one HP interaction and two PP interactions.

An HP protein configuration can be represented as a walk in the lattice (sequence of moves). In the sequence of moves, the two initial residues are located adjacent in the lattice. Each of the other residues is located to the left, to the right, or forming a line with the previous two residues. For a given HP sequence and lattice, X_i will represent the relative move of residue i in relation to the previous two residues. Taking as a reference the location of the previous two residues in

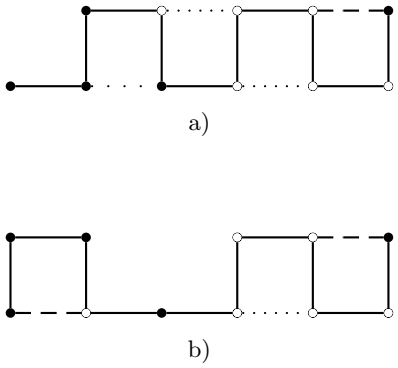

Figure 3: (a): One possible configuration of sequence $HHHPHPPPPPH$ in the HP functional model. Hydrophobic proteins are represented by black beads and polar proteins, by white beads. There is one HH interaction (represented by a dotted line with wide spaces), one HP interaction (represented by a dashed line) and two PP interactions (represented by dotted lines) contacts. (b): Another possible configuration of the same sequence with a different pattern of interactions.

the lattice, X_i takes values in $\{0, 1, 2\}$. With respect to the location of the previous two residues, $x_i = 0$ means that residue i is located to left, similarly $x_i = 1$ and $x_i = 2$ respectively mean that residue i will be located in line with the previous two residues and to their right. Values for X_1 and X_2 are meaningless, they are arbitrarily set to 0. This codification is called relative encoding [12]. The representations of configurations in Figure 3 a) and b) are $\mathbf{x}^i = (0, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0)$ and $\mathbf{x}^j = (0, 0, 2, 2, 0, 1, 0, 2, 2, 0, 0)$, respectively.

6. EXPERIMENTS

In this section we evaluate the behavior of the introduced EDAs. First, we introduce the function benchmark and the parameters used by the algorithms. Then, we explain how the experiments were designed. Finally, the results of the experiments are presented.

6.1 Function benchmark and parameters of the algorithms

Table 1 shows the HP instances used in our experiments. The values shown in Table 1 correspond to the best-known solutions ($H(x^*)$) for the 2-d regular lattice.

The parameters of the EDAs have been set as follows. Truncation selection with parameter T has been used. In this selection scheme, the best $T \cdot N$ individuals of the population are selected to construct the probabilistic model. We apply a replacement strategy called best elitism in which the selected population at generation g is incorporated into the population of generation $g + 1$, keeping the best individuals found so far and avoiding to revaluate their fitness function. The algorithm stops when the maximum number of generations is reached. In all the experiments we use a population size of $N = 4\dot{n}$ and 500 generations.

To compare the results of the EDAs, we conducted 15 experiments for each HP instance and algorithm. A total number of $9 \times 9 \times 15 = 1215$ experiments were conducted.

Table 1: HP instances used in the experiments. The search space of each instance is 2^n where n is the size of the instance.

inst.	n	$H(\mathbf{x}^*)$	sequence
$s1$	20	-9	$\{HP\}^2\{PHH\}^2PHPHHPPHPH$
$s2$	24	-9	$HH(PPH)^6H$
$s3$	25	-8	$PPHPPHHP^4HHP^4HHP^4HH$
$s4$	36	-14	$P^3\{H^2P^2\}^2P^3H^7P^2H^2P^3\{PH^2\}^2P^2$
$s5$	48	-23	$PPHPPHHPPHHP^5H^{10}P^6$ $HHPPHHPPHPPH^5$
$s6$	50	-21	$HHPHPHPHPH^4PHP^3HP^3HP^4$ $HP^3HP^3HPH^4\{PH\}^4H$
$s7$	60	-36	$PPH^3PH^8P^3H^{10}PHP^3$ $H^{12}P^4H^6PHHPHP$
$s8$	64	-42	$H^{12}\{PH\}^2\{P^2H^2\}^2PPH\{P^2H^2\}^2$ $PPH\{PPHH\}^2PPHPHPH^{12}$
$s9$	85	-53	$H^4P^4H^{12}P^6H^{12}P^3H^{12}P^3$ $H^{12}P^3HP^2H^2P^2H^2P^2HPH$

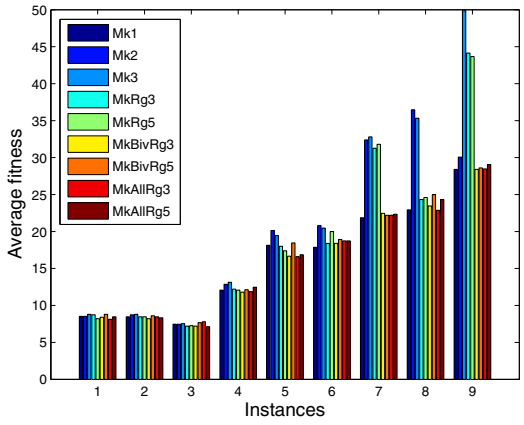

Figure 4: Results achieved by the 9 EDAs in all the instances.

The performance of the algorithms was evaluated considering the average best fitness obtained in the 15 experiments. We do not compare the algorithms with respect to the best known solutions because these optimal solutions have been found using an unaffordable number of evaluations and/or the intense application of local optimization procedures.

6.2 Numerical results

Our first objective was to evaluate the global behavior of the EDAs that used regularized models. The main question to answer is whether EDAs that use regularized models can outperform the k-order Markov EDAs that use higher order interactions. Other questions to discern are:

1. To what extent the regularized models that incorporate bivariate interactions can outperform the simpler regularized models?

2. How do the EDAs that use regularized model rank

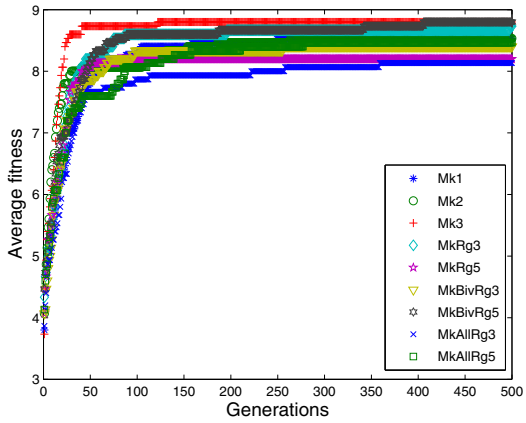

Figure 5: Average fitness at each generation of the EDAs for instance s1.

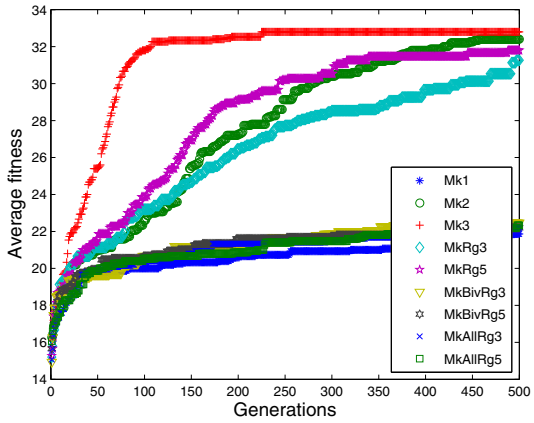

Figure 6: Average fitness at each generation of the EDAs for instance s7.

in comparison to Mk1EDA, the EDA that uses the simplest model among all compared?

3. Do the results achieved by the EDAs that use regularized models scale with the number of variables?

4. Do the regularized models capture any type of relevant information about the problem?

Figure 4 shows the average best fitness achieved by all the algorithms for all the instances. For instances $s1$, $s2$ and $s3$, it is difficult to find clear differences between the algorithms. These are the easiest problems and good results can be achieved by all the algorithms. Notice however, that the more complex models used by MkEDA_2 and MkEDA_3 do not produce an important gain in the results. For instances $s4, \ldots, s8$, results achieved by MkEDA_2 and MkEDA_3 are superior to the other algorithms. This is not surprising since these algorithms are able to capture more information about the problems. However, the results achieved by the MkRg3EDA and MkRg5EDA algorithms for instances $s6$, $s7$ and $s9$ are remarkable. For all these instances the algorithms obtain an improvement over (MkEDA_1). For instance $s6$, the behavior of algorithm MkRg5EDA is very close to MkEDA_2 and MkEDA_3. For the largest instance ($s9$), MkRg3EDA and MkRg5EDA outperform MkEDA_2. Finally, for instance $s7$, the algorithms are also close to MkEDA_2 and MkEDA_3.

We take a closer look to the behavior of the algorithms by analyzing the average fitness of the population at each generation. These results, for instances $s1$, $s7$, $s8$ and $s9$, are shown in Figures 5-8. We have chosen one of easiest problems and the three largest instances to analyze different scenarios of the algorithms behavior.

The results shown in Figure 5 for instance $s1$ suggest that, although the results for all the EDAs are similar, Mk3EDA is able to reach an early advantage over the other algorithms. It converges earlier to better solutions. However, as shown, in Figure 7, in some cases Mk2EDA reaches better solutions than Mk3EDA in later generations. Figures 6 and Figure 8 also illustrate how MkRg3EDA and MkRg5EDA achieve their good results for instances $s7$ and $s9$. They start slower than Mk3EDA but keep the improvement of the fitness at a better pace. This is particularly evident for instance $s7$ for which the curves shown in Figure 6 indicate that algorithms MkRg3EDA and MkRg5EDA could still improve their results if more generations were allowed.

The results shown in Figures 4-8 help to answer some of the questions posed at the beginning of this section. The answer to the main question is negative. At least for the instances considered, the EDAs that use regularized models can not outperform EDAs that use models of higher complexity. However, it is not clear whether these results could change if more difficult instances were tested or a higher number of generations were allowed.

The regularized models that incorporate bivariate interactions do not seem to produce a gain in the quality of the results achieved by MkRg3EDA and MkRg5EDA. More expressive models do not necessarily improve the optimization results in EDAs due to issues like overfitting. The Markov EDAs that use the linear regularized models clearly outperform the Mk1EDA for the largest instances ($s6$, $s7$, $s8$ and $s9$). For the smaller instances, there is no a clear winner. This means that the regressed models capture some sort of relevant information about the interactions between the variables and that this information is useful for the search.

We cannot affirm that the results achieved by MkRg3EDA and MkRg5EDA scale with the number of variables since we have not included a large number of instances in our experiments. However, as noticed above, the best results achieved by these algorithms are achieved for the largest instance. Therefore, at least for the examples studied, the results of the algorithms do not deteriorate with the number of variables. It is important to note, that the population size used by the EDAs has been set according to the number of variables of each problems, i.e. $N = 4n$.

To answer the final question, we execute a new run of MkRg3EDA, saving the regularized models learned in each generation for each variable. The idea is to inspect the coefficients corresponding to each of the previous k values in order to identify which of the previous variables are "the most informative" for the model. We do not take into consideration the sign of the coefficients. Figure 9 shows the results of

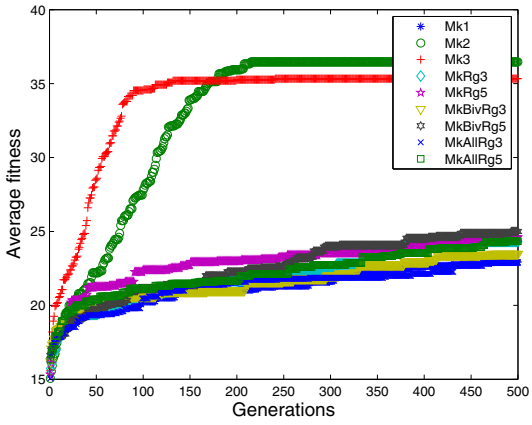

Figure 7: Average fitness at each generation of the EDAs for instance s8.

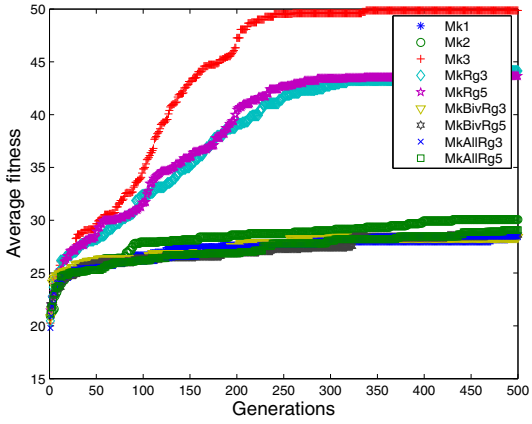

Figure 8: Average fitness at each generation of the EDAs for instance s9.

the experiment. In Figure 9, there are 3 coefficients for each of the k variables. This is so because the multi-log model uses three loglineal models to make the prediction. The first group of three coefficients correspond to the variable that is further away from variable X_i, i.e. X_{i-k}, the second group corresponds to X_{i-2} and the last group of coefficients correspond to variable X_{i-1}. Light color correspond to greater coefficient values. It can be seen that darker colors are concentrated in the columns corresponding to the first group of coefficients. This fact is more or less consistent for all the variables. This seems to indicate that as variables are further away from X_i their contribution to the prediction is less critical. The analysis of the coefficients could also be used to "prune" the k-order Markov models, computing a single value k_i for each variable X_i. This way, only the most informative k_i values would be learned for each variable X_i.

7. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed the use of regularized probabilistic graphical models to investigate the influence

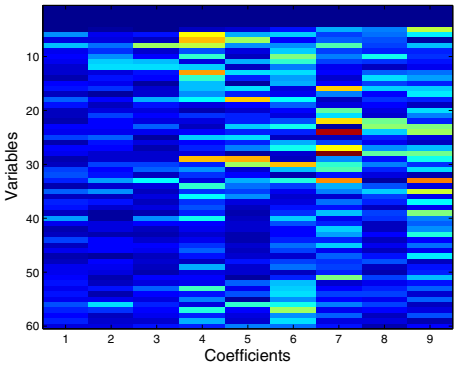

Figure 9: Coefficients associated to each of the k variables in the multinomial regularized model learned for algorithm MkRg3EDA.

of the complexity of k-order Markov models in the behavior of EDAs that use these models. We have analyzed three variants of the regularized models that comprise single independent variable contributions, bivariate contributions and the combination of these two models. Our results show that EDAs that use regularized models based on linear combinations of independent variables can improve the results of Mk1EDA for difficult problems. They can also approximate the results of MkEDAs for $k = 2$ and $k = 3$. However, in none of the problems considered, the introduced regularized k-order Markov EDAs were able to outperform MkEDA_3, the EDA that uses the higher order distributions. Experiments using other functions benchmarks are needed to fully validate the introduced algorithms.

One of the main contributions of the approach introduced in this paper is that it allows to investigate what happens to the performance of EDAs when the accuracy of the model estimation is relaxed using different types of approximations. We have shown that the regularized models can support information about the role played by the previous k variables in the Markov approximation. This information could be used to refine the models.

Learning a regularized model implies to solve a convex optimization problem. This represents an additional cost for EDAs which is increased because for each variable, a different model is learned. Certainly, using a more complex phase of parameter estimation, the benefits of avoiding structural learning may deteriorate. However, there exist many complex problems where the cost of the fitness evaluation justifies the use of more costly, but still feasible, estimation and sampling techniques. Furthermore, other alternative, less computationally expensive, ways of using regularization have been recently investigated and could be tested in the context of continuous k-order Markov models [11].

The regularized EDAs introduced in this paper can also be seen as a particular way of hybridizing EDAs with other traditional optimization algorithms, one of the areas where research on EDAs seems to be more promising [21]. Future research lines could include tuning the regularization parameter to improve accuracy in the approximation and developing a flexible k-order model able to identify k according to the nature of the problem.

8. ACKNOWLEDGMENTS

This work has been partially supported by the TIN2010-20900-C04-04, Consolider Ingenio 2010 - CSD2007-00018 projects (Spanish Ministry of Science and Innovation) and the CajalBlueBrain project.

9. REFERENCES

[1] B. Berger and T. Leight. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

[2] B. Chen and J. Hu. A novel clustering based niching EDA for protein folding. In *Proceedings of the World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009*, pages 748–753. IEEE, 2010.

[3] B. Chen, L. Li, and J. Hu. An improved backtracking method for EDAs based protein folding. In *Proceedings of ICROS-SICE International Joint Conference 2009*, pages 4669–4674, 2009.

[4] C. Cotta. Protein structure prediction using evolutionary algorithms hybridized with backtracking. In J. Mira and J. R. Alvarez, editors, *Artificial Neural Nets Problem Solving Methods*, volume 2687 of *Lecture Notes in Computer Science*, pages 321–328. Springer, 2003.

[5] P. Crescenzi, D. Goldman, C. H. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. *Journal of Computational Biology*, 5(3):423–466, 1998.

[6] J. S. De Bonet, C. L. Isbell, and P. Viola. MIMIC: Finding optima by estimating probability densities. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 424–430. The MIT Press, Cambridge, 1997.

[7] K. A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6):1501–1509, 1985.

[8] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning. Data Mining, Inference and Prediction*. Springer Series in Statistics. Springer, 2001.

[10] J. D. Hirst. The evolutionary landscape of functional model proteins. *Protein Engineering*, 12:721–726, 1999.

[11] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga. Regularized model learning in estimation of distribution algorithms for continuous optimization problems. Technical Report UPM-FI/DIA/2011-1, Department of Artificial Intelligence, Faculty of Informatics, Technical University of Madrid, January 2011.

[12] N. Krasnogor, B. P. Blackburne, E. K. Burke, and J. D. Hirst. Algorithms for protein structure prediction. In *Parallel Problem Solving from Nature - PPSN VII*, volume 2439 of *Lecture Notes in Computer Science*, pages 769–778. Springer, 2002.

[13] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

[14] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

[15] H. Mühlenbein and G. Paaß. 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*, volume 1141 of *Lectures Notes in Computer Science*, pages 178–187, Berlin, 1996. Springer.

[16] M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[17] R. Santana. A Markov network based factorized distribution algorithm for optimization. In *Proceedings of the 14th European Conference on Machine Learning (ECML-PKDD 2003)*, volume 2837 of *Lecture Notes in Artificial Intelligence*, pages 337–348, Dubrovnik, Croatia, 2003. Springer.

[18] R. Santana, P. Larrañaga, and J. A. Lozano. Protein folding in 2-dimensional lattices with estimation of distribution algorithms. In *Proceedings of the First International Symposium on Biological and Medical Data Analysis*, volume 3337 of *Lecture Notes in Computer Science*, pages 388–398, Barcelona, 2004. Springer.

[19] R. Santana, P. Larrañaga, and J. A. Lozano. Component weighting functions for adaptive search with EDAs. In *Proceedings of the 2008 Congress on Evolutionary Computation CEC-2008*, pages 4067–4074, Hong Kong, 2008. IEEE Press.

[20] R. Santana, P. Larrañaga, and J. A. Lozano. Protein folding in simplified models with estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation*, 12(4):418–438, 2008.

[21] R. Santana, P. Larrañaga, and J. A. Lozano. Research topics on discrete estimation of distribution algorithms. *Memetic Computing*, 1(1):35–54, 2009.

[22] S. Shakya and J. McCall. Optimization by estimation of distribution with DEUM framework based on Markov random fields. *International Journal of Automation and Computing*, 4(3):262–272, 2007.

[23] J. Yang, H. Xu, Y. Cai, and P. Jia. Effective structure learning for EDA via l1-regularized Bayesian networks. In *GECCO-2010: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pages 327–334, New York, NY, USA, 2010. ACM.

[24] J. Zhu and T. Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3):427, 2004.

[25] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.