Introducing ℓ_1-regularized Logistic Regression in Markov Networks based EDAs

Malagò Luigi
Politecnico di Milano
Via Ponzio, 34/5
20133 Milano, Italy
Email: malago@elet.polimi.it

Matteucci Matteo
Politecnico di Milano
Via Ponzio, 34/5
20133 Milano, Italy
Email: matteucci@elet.polimi.it

Valentini Gabriele
Politecnico di Milano
Via Ponzio, 34/5
20133 Milano, Italy
Email: gabriele.valentini@mail.polimi.it

***Abstract*—Estimation of Distribution Algorithms evolve populations of candidate solutions to an optimization problem by introducing a statistical model, and by replacing classical variation operators of Genetic Algorithms with statistical operators, such as estimation and sampling. The choice of the model plays a key role in the evolutionary process, indeed it strongly affects the convergence to the global optimum. From this point of view, in a black-box context, especially when the interactions among variables in the objective function are sparse, it becomes fundamental for an EDA to choose the right model, able to encode such correlations. In this paper we focus on EDAs based on undirected graphical models, such as Markov Networks. To learn the topology of the graph we apply a sparse method based on ℓ_1-regularized logistic regression, which has been demonstrated to be efficient in the high-dimensional case, i.e., when the number of observations is much smaller than the sample space. We propose a new algorithm within the DEUM framework, called DEUM$_{\ell_1}$, able to learn the interactions structure of the problem without the need of prior knowledge, and we compare its performance with other popular EDAs, over a set of well known benchmarks.**

I. Introduction

Estimation of Distribution Algorithms [1] are a well known paradigm of optimization in the Evolutionary Algorithms community. EDAs evolve a population of promising solutions, from one generation to the next, by sampling the offspring from a probabilistic model of the population. Most of the EDAs use Probabilistic Graphical Models (PGMs) in order to factorize the joint probability distribution associated to the variables in objective function, and usually in the literature are categorized according to the complexity of the statistical model employed, i.e., the maximum order of interactions among the variables that the model is able to encode. Otherwise, EDAs can be divided into two main categories, those which employ undirected models that include Markov Networks (MNs), also known as Markov Random Fields (MRFs) [2] or log-linear models, and those based on Directed Acyclic Graphs (DAGs) such as Bayesian Networks (BNs). FDA [3], MN-EDA [4] and the algorithms based on the DEUM framework [5] are examples of multivariate EDAs which belongs to the former case, while BOA [6] and hBOA [7] to the latter.

Probably one of the most important distinction, especially when solving black-box problems, is whether the statistical model is determined a priori, or if it is learned at runtime. This distinction is very important, indeed the choice of a good model is one of the main factors in order to increase the probability of success of the algorithm. The notion of good model must be defined more precisely, in particular, it can be expressed in terms of the relation between the correlations present in the function to be optimized and those that can be captured by the conditional independence structure of a statistical model.

One of the great advantages of introducing statistical operators in the design of an evolutionary algorithm, such as model building, model estimation and sampling, is given by existence of a vast literature in Statistics and Machine Learning about PGMs in the high-dimensional context, that is, when the number of observations is much smaller compared to the sample space, which is exactly the context for an EDA.

In the context of BNs, different popular model selection criteria for DAGs have been applied to EDAs, see for instance the use of the Bayesian Information Criterion (BIC) or the Bayes Dirichlet equivalent scoring (BDe) in BOA [6]. On the other side, dealing with MNs, classical approaches include hypothesis test for conditional independence and measures of correlation, such as those based on Cross Entropy. Most of these techniques have been implemented and tested in DEUM [8], [9], a meta-heuristic that belongs to the broad class of EDAs which employs MNs to model the interactions among variables.

In this paper we focus on undirected graphical models, and when dealing with MNs we distinguish between model selection, that is the learning of a statistical model that determines which interactions are present, or, in other words, the topology of the PGM, and model fitting, i.e., the estimation of the parameters of the model. Even if usually in MRFs only interactions of order two are considered, the saturated model, i.e., a model able to represent any possible interaction among the variables, requires an exponential number of terms.

Good models are those able to capture most of the interactions encoded in the objective function, since in this way the number of local solutions is reduced, cf. [10]. On the other side, since the greater the number of interactions is, the less tractable the model becomes, cf. [11], model selection makes much more sense if the function is sparse, that is, only a restricted number of interactions appear in the expansion

of the fitness function. Such hypothesis applies in many real world problems, see for example the Ising spin glass based on regular lattices and MAXSAT, which are hard to solve, indeed the minimizing of a quadratic function is NP-hard in the general formulation, also under the hypothesis of sparse interactions.

Motivated by these observations, we propose to apply an ℓ_1-regularized method to determine the structure of conditional independences of a MRF. Such approach to model selection is based on the idea of performing learning over the conditional distribution rather than on the joint distribution, and then apply a penalized method to solve each of the logistic regression problems obtained. In contrast to other model selection approaches employed in EDAs based on MNs, our method does not require control parameters, which in general may depend from the particular problem, such as an upper bound on the edge cardinality in the graphical model.

To the best knowledge of the authors, the use of ℓ_1-regularized logistic regression first appeared in the contest of EDAs in L1BOA [12], where the method is employed to select a subset of candidate parents for each node in a BNs by minimizing the MDL, and to limit the number of candidate models, before a greedy search based on classical Bayesian scoring criteria determines the final topology of the DAG. By contrast, we directly apply logistic regression to estimate the statistical model, for which the parameters are then estimated. The use of regularized models also appears in [13], in the EDA literature for continuous optimization.

The paper is organized as follows. In Section II we review the main features of the algorithms that belong to the DEUM framework, while in Section III we introduce ℓ_1-regularized logistic regression in the context of model selection for the edge set of a MRFs. Section IV includes a detailed description of DEUM$_{\ell_1}$, an EDA which implements model selection by solving a set of logistic regression problems, penalized by ℓ_1 norm, and in Section V we present the experimental results of DEUM$_{\ell_1}$ over a set of benchmarks of increasing difficulty, and compare the performance of the algorithm with other popular EDAs. Finally, we conclude in Section VI.

II. The DEUM Framework

In this section we introduce the DEUM framework, a paradigm to optimization based on Markov Networks, see [5]. This meta-heuristic belongs to the broad class of EDAs and more in general can be classified among the model-based algorithms, since candidate solutions to the optimization problem are generated by sampling from a joint probability distribution. Given a statistical model, at each iteration the parameters of a probability distribution are estimated from a sample of selected solutions, so that each run of the algorithm can be described as a sequence of in a statistical model within the probability simplex. The DEUM framework differs from other multivariate EDAs, such as BOA [6], since it employs MNs, whose conditional independence structure among variables can be represented by a statistical model that belongs to the exponential family [14].

In the following we restrict our attention to the class of real-valued function $f : \Omega = \{+1, -1\}^n \to \mathbb{R}$, defined over a vector of binary variables $x = (x_1, \ldots, x_n)$. This function are also know as pseudo-Boolean functions [15]. We employ the harmonic encoding[1] for binary variables, rather than the usual $0/1$, so that any f can be uniquely expressed as a square-free polynomial

$$f(x) = \sum_{\alpha \in I} c_\alpha x^\alpha, \tag{1}$$

where $\alpha = (\alpha_1, \ldots, \alpha_n) \in L = \{0, 1\}^n$, since $x_i^2 = 1$, and $x^\alpha = \prod_{i=1}^n x_i^{\alpha_i}$.

We approach the problem of finding the maximum of f by sampling from a sequence of probability distributions that we would like to converge in probability to the $\delta(x)$ distribution that concentrate all probability mass over optimal solutions.

The DEUM framework employs statistical models of the exponential family

$$p(x; \theta) = \exp\left(\sum_{i=1}^{k} \theta_i T_i(x) - \psi(\theta)\right), \quad \theta \in \mathbb{R}^k, \tag{2}$$

where T_i are the sufficient statistics, which are pseudo-Boolean functions themselves, and $\psi(\theta)$ is the normalizing factor. The sufficient statistics of the model correspond to square-free monomials representing the α-monomial interactions between the variables. Equation (2) can be written as

$$\log p(x; \theta) = \sum_{\alpha \in L} \theta_\alpha x^\alpha - \psi(\theta),$$

so that the correspondence between log-linear models and MNs becomes clear. Such family of distributions is a good candidate model from a theoretical point of view, indeed it includes the Gibbs (or Boltzmann) distribution, a one dimension model which is known to converge to the $\delta(x)$ distribution over the maxima of f for β that goes to infinity, see for instance [16]. More in general, it has been proved in [10] that if f belongs to the span of the sufficient statistics of the exponential model, then the gradient of the expected valued of f, which is a continuous function defined over the model, never vanishes. Moreover, we have that from any distribution q in the statistical model, there exits a curve that follows the direction of maximum decrement of $\mathbb{E}[f]$ and converges to the $\delta(x)$ distribution over the minima of f, given by

$$p(x; \beta) = \frac{q e^{\beta f}}{\mathbb{E}_q[e^{\beta f}]}, \quad \beta \in \mathbb{R}.$$

Informally, this implies that if the model selected by an EDA is a good model, i.e., it encodes the right interactions present in the function, then the algorithm has greater probabilities to converge to the optimum.

EDAs based on MRFs and in general on the exponential family in Equation (2) have to deal with the three main issues,

[1]The harmonic encoding is employed to ensure mathematical symmetry between possible allele values. This is a standard practice when dealing with MNs.

how to choose a model, how to estimate its parameters, and how to sample from a probability distribution.

As to the model fitting, it is a common approach in the EDA literature for most of the algorithms based on Directed Acyclic Graphs (DAG) to perform maximum likelihood estimation. In the context of MRF, in the general cases, this implies the solution of the system of equations $\mathbb{E}_{\hat{\theta}}[X^\alpha] = \hat{\mathbb{E}}[x^\alpha]$, which is not computationally feasible with no assumptions of the structure of the conditional independences of the model. As a consequence in order to make the estimation of the parameters tractable, many algorithms employ a factorization of the joint probability distribution based on cliques of the associated MRF, while others work with junction trees or junction graphs. The approach developed in DEUM differs from others EDAs, since maximum likelihood estimation is replaced by an explicit modelling of the fitness function, as described in detail in the following.

A. The MRF Fitness Model

The basic assumption behind the MRF Fitness Model (MFM) [17] employed by DEUM is that a good probability distribution for sampling is given by

$$p(x) = \frac{f(x)}{Z}, \quad Z = \sum_{y \in \Omega} f(y), \tag{3}$$

where the probability of each point is proportional to its fitness value. The approach is similar to the choice of the Gibbs distribution, even if Equation (3) does not belong to the exponential family. Given a set of monomials $M \subset L$ which form a basis for the energy function

$$U(x) = \sum_{\alpha \in M} \theta_\alpha x^\alpha,$$

or equivalently by specifying a set of interactions among the variables represented in a MRF, the MFM identifies a probabilistic model for the fitness function and estimates the parameters of probability distribution in the exponential model

$$p(x) = \frac{e^{-U(x)/T}}{\sum_{y \in \Omega} e^{-U(y)/T}} = \frac{f(x)}{\sum_{y \in \Omega} f(y)}$$

by solving the following system of linear equations

$$\sum_{\alpha \in M} \theta_\alpha x^\alpha = -\ln f(x), \tag{4}$$

where T is assumed to be 1 for convenience. Since the logarithmic function is monotonic, it is possible to find the maximum of $f(x)$ by minimizing $U(x)$.

The number k of variables in θ to be determined by solving the system depends on the choice of the sufficient statistics of the exponential model, or equivalently on the edge set in the MRF. The set of monomials M can be fixed a priori, as in IsDEUM [18], an algorithm explicitly designed to find the ground states of 2D spin glass problems [2], or can be inferred from the sample at runtime.

In both cases the parameters in θ are determined by solving a linear system, in which each equation satisfying the MFM is given by an individual in the population. More formally, the vector of parameters θ is determined by solving

$$F = A\theta^T, \tag{5}$$

where F is a column vector containing the $-\ln(f(x))$ for all elements in the selected population[2], and A is the matrix of the monomial expansion of $U(x)$ in (4). In doing so, we approximate the parameters θ by fitting the MFM to the selected population.

B. Model Selection

In the DEUM framework, besides the algorithms where the structure of the statistical model employed is fixed, recently a number of different approaches to structure learning have been proposed. For instance DEUM-LDA [8] is based on linkage discovery by probing the fitness function [19]. LDA is a well known algorithm able to determine the monomials that appear in the expansion of f, provided a maximum order k of interactions. Such approach does not require a population of solutions, and it can be applied only to determine the model once, before running an EDA. However, recovering the topology of the MN through LDA requires an overall amount of $\binom{n}{k} \cdot 2^k$ fitness calls.

On the other side, classical statistical hypothesis tests can be employed to infer the conditional independence structure from the current population. For instance, this is the case of the Pearson's χ^2 statistics employed in DEUM$-\chi^2$ [8]. Other approaches have implemented in DEUM, for example, in [9] the Cross Entropy information and the JEMP critera, which are respectively measures of randomness and correlation among variables, allow to determine the set of more relevant interactions provided some threshold values.

III. ℓ_1-REGULARIZED LOGISTIC REGRESSION

In this section we present an approach based on ℓ_1-logistic regression [20] to learn a statistical model from the exponential family in Equation (2), given a sample of observations, which in our case corresponds to a population of selected individuals. The main idea is to infer the conditional independence structure of the MN by recovering the neighbourhood of each variable. Indeed, as for sampling, solving the MLE problem for the joint probability distribution is computationally infeasible for the exponential family in the general case, since it involves the computation of the normalizing constant $\psi(\theta)$. Instead, the conditional probability distribution of each variable given the others does not depend on the normalizing constant, so that it is possible to apply MLE and to retrieve the topology of the MRF by solving n logistic regression problems, one for each variable.

Let $x_{\setminus i}$ be the vector of all variables in x except x_i, similarly $L_i \subset L$ the set of α such that $\alpha_i = 1$ and θ_i the vector of the parameters θ corresponding to α_i, and $x^{\alpha \setminus i}$ the α-monomial with $x_i^{\alpha_i} = 1$. By simple calculations, it easy to verify that

[2]In order to consistently estimate the parameters θ the matrix A must be invertible.

the conditional distribution of the random variable X_i given all the other variables becomes

$$p_i(x_i|x_{\setminus i};\theta_i) = \frac{\exp\left(2x_i \sum_{\alpha\in L_i}\theta_{\alpha\setminus i}x^{\alpha\setminus i}\right)}{\exp\left(2x_i \sum_{\alpha\in L_i}\theta_{\alpha\setminus i}x^{\alpha\setminus i}\right)+1}, \quad (6)$$

so that X_i can be interpreted as the response variable in a logistic regression problem. Under the hypothesis of sparsity, the method employed to estimate θ_i is based on ℓ_1-regularized logistic regression of X_i, given a collection $\mathcal{P}$ of m i.i.d. observations $\{x^{(1)},\dots,x^{(m)}\}$. In particular, we need to solve the following convex program

$$\min_{\theta_i}\left(\mathcal{L}(\theta_i|\mathcal{P}) + \lambda||\theta_i||_1\right),$$

where $\mathcal{L}(\theta_i|\mathcal{P})$ is the negative log-likelihood, and $\lambda > 0$ is the regularization parameter. The solution of the n regression problems return a set of sparse vectors $\hat{\theta}_i$ from which we can infer the set of sufficient statistics of the exponential model.

In the most general case θ_i has 2^{n-1} components, and the logistic regression problem remains intractable. As a consequence, to solve the minimization problem in polynomial time, the maximum degree of interactions of the α monomials is limited, usually to set to the second order, so that each θ_i has $n-1$ components.

The sparsity pattern of θ_i strongly depends on the value of the regularizing parameter λ. For $\lambda \to \infty$, all θ_i will vanish, while for $\lambda \to 0$, the solution will equal the MLE, which is not sparse in general. As a consequence, the value of λ must be chosen carefully, according to the number of variables n, the sample size m, and the density d of the MN. In [20], Ravikumar et al. studied the problem of estimating the topology of a MRF with binary variables using ℓ_1-regularized logistic regression. Their main result, which applies in our context, provides conditions under which the topology of the MN can be correctly reconstructed. In particular under some assumptions on the structure of the logistic regression problem, they prooved the network can be efficiently reconstructed if the regularization parameter equals

$$\lambda = \sqrt{\frac{\log n}{m}}, \quad (7)$$

where m depends on the type and density of the graph. Such theoretical result is obtained under the hypothesis that the sample is i.i.d. from to an unknown probability distribution associated to the fitness function. Usually such hypothesis cannot be satisfied in black-box optimization, since it would imply to be able to sample from

$$p(x) = \frac{e^{f(x)}}{Z}, \quad Z = \sum_{y\in\Omega} e^{f(y)}, \quad (8)$$

which in turns could be done efficiently for example if the interactions of f would be known. In order to deal with this issue, we propose to perform ℓ_1-regularized logistic regression over a subset of the population, selected according to the fitness value of the individuals. This procedure can only approximates an i.i.d. sample, but from our experiments we were able to correctly reconstruct the topology of the MN by selecting individuals from a randomly generated initial population.

IV. The DEUM$_{\ell_1}$ Algorithm

In this section we present DEUM$_{\ell_1}$, a fully multivariate DEUM algorithm that employs an ℓ_1 norm for model selection. The algorithm is able to learn the interactions present in the fitness function and encode them in a MN in black-box contexts, when no information about f is available.

Algorithm 1: DEUM$_{\ell_1}$

```
Set t = 0
Randomly generate a population P^t
while Stopping criteria are not met do
    Evaluate fitness of the individuals in P^t
    Select a subset P_s^t from P^t
    Learn the statistical model M^t from P_s^t
    Estimate the parameters θ̂^t of M^t from P_s^t
    Sample P^{t+1} using the Gibbs sampler
    Set t = t + 1
```

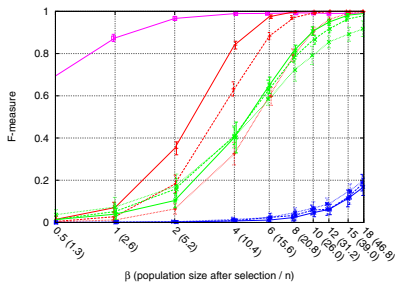
Algorithm 1 shows the main steps of DEUM$_{\ell_1}$. First a random population $\mathcal{P}^0$ of candidate solutions is generated, then the algorithm proceeds iteratively until stopping criteria are satisfied, for instance when maximum number of generations is reached or population reached convergence. The main loop of the algorithm consists of evaluation of the fitness f for the individuals in the population, and the selection of a subset of the population to be employed in model selection and parameter estimation. Next, the selected sample $\mathcal{P}_s^t$ is used to learn which are the relevant interactions in the function with an algorithm based on ℓ_1-regularized logistic regression which will be described later. Afterward, the value of $\hat{\theta}$ is estimated by solving the linear system in Equation (5). Finally, a new population is generated by sampling from the new estimated distribution.

A. Learning the Undirected Structure

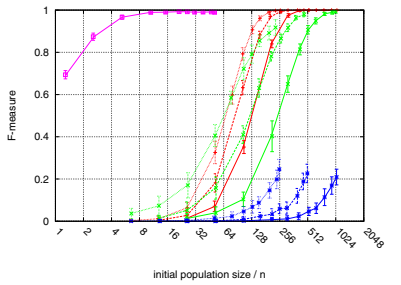
As introduced in Section III, the task of learning the structure of a Markov Network can be reduced to recovering the sparsity pattern of the regression vector in a series of logistic regression problems. The overall procedure is presented in Algorithm 2.

Algorithm 2: ℓ_1-MN Learning

```
Set M as a MN with no interactions
Set λ = sqrt(log n / m)
for i ← 1 to n do
    Set X_i as response variable
    Set X_{\i} as covariates
    Solve the logistic regression in Eq. (6) of X_i over X_{\i}
    foreach j ∈ {1,...,n} \ i do
        if j-th component of θ̂_i ≠ 0 then
            Add the edge (i, j) to M
return M
```



(a)



(b)

Fig. 1. Evaluation of the performance of ℓ_1-regularized logistic regression applied to recover the topology associated to a 8×8 2D spin glass network, with ± 1 coefficients. The color of the line identifies different initial samples resulting from different selection policies. Red: truncation selection; green: Boltzmann selection with $T = 1/0.225$; blue: binary tournament selection. Solid line $ps = 5\%$, dashed line $ps = 10\%$, dotted line $ps = 20\%$. Purple line, i.i.d. sample generated through Gibbs sampler. Experiments average over 30 executions. (a) X axis is the control parameter β (and population size after selection) (b) X axis is the population size of DEUM$_{\ell_1}$ before selection.

The ℓ_1-MN Learning algorithm begins by initializing a network $\mathcal{M}$ with no edges between the nodes. Then, for each variable X_i in the optimization problem, a ℓ_1-regularized logistic regression problem is solved in which the variables $X_{\setminus i}$ play the role of covariates. The maximum order of interactions in the Markov Network is set to two, and each logistic regression problem is resolved over a set of $n-1$ covariates. In particular we used an efficient interior point method to solve the ℓ_1-logistic regression problem originally proposed in [21]. Once the logistic regression problems are solved, and the sparsity pattern $\hat{\theta}_i$ are recovered, for each j-th component in $\hat{\theta}_i \neq 0$, the edge (i, j) is added to $\mathcal{M}$.

Due to the symmetry of the correlations, it is possible to first recover the neighbourhood for X_1, and fix the interactions obtained, and then solve the logistic regression for X_2, over the remaining $n-2$ variables. Similarly for X_3, given the neighbourhood of X_1 and X_2, and so on. This results in a reduction of the number of covariates taken into account in each logistic regression problem, and in a overall reduction of the computational cost of the algorithm. In our experiments, we found out that the structure of interactions recovered with both the complete and the reduced approach is almost the same, up to the finite numerical precision of the implementation.

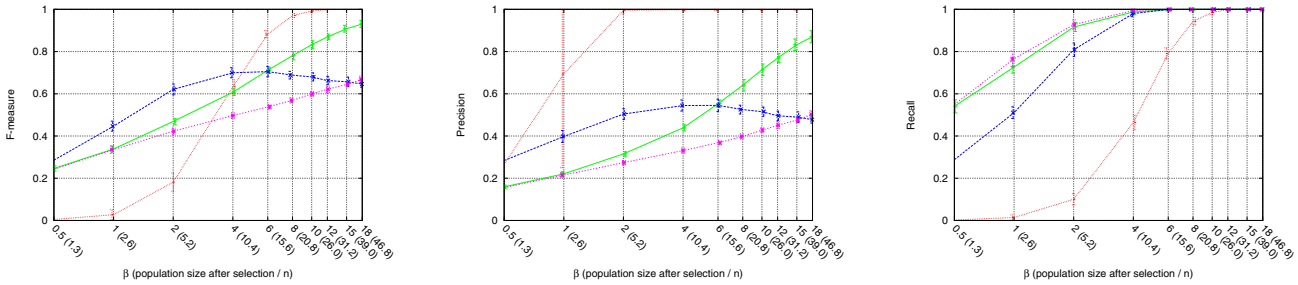
As described in the previous section, in order to successfully recover the topology of the network, and dimension the regularization parameter according to Equation (7), an i.i.d. sample is required. In order to approximate such sample, we propose to run the learning algorithm over a restricted subset of the population, chosen according to a selection procedure based on the value of the fitness function. We tested different selection policies, such as Boltzmann selection [22], with temperature parameter T, truncation selection, with percentage of selection equal to ps, and binary tournament selection. We evaluated the selection policies starting from a random sample generated from the uniform distribution, and we compared the performance with respect to an i.i.d. sample of the same size, generated from the distribution in Equation (8). As in [20], we performed the experiments for different population sizes m, scaling m as $m = 10\beta d \log n$, where $\beta > 0$ is a control parameter which depends on the graph type, and d the expected maximum number of interactions for a node in the MN. Figure 1 shows the results, for a 2D spin glass lattice, of 8×8 variables, with coefficients ± 1, with $d = 4$. We evaluated the performance by means of the F1-score, the harmonic mean of precision, the percentage of the edge learned that are originally in the unknown model, and recall, the portion of interactions originally present in the model that have been found.

The algorithm applied to the i.i.d. sample shows the best performances as expected, and behaves accordingly to results presented in [20]. Among the other selection operators evaluated, we see that truncation selection performs better than Boltzmann selection, that is, it requires smaller values of β in order to maximize F1-score, see Figure 1(a). Similarly we notice that lower selection thresholds increase the performance of model selection, nevertheless resulting in larger initial populations. Figure 1(b) shows the same results as a function of the initial population before selection.

Finally, we compared the performance of the ℓ_1-regularized logistic regression method with other model selection techniques recently implemented in DEUM, such as CE, χ^2, and JEMP. Figure 2 shows that given the same sample size, ℓ_1-regularized logistic regression outperforms the other techniques, since requires smaller populations to correctly detect the topology of the network in the case of the 8×8 2D lattice of a random generated ± 1 spin glass model. Moreover notice that for small values of β, differently form the other methods, with ℓ_1-regularized logistic regression we have higher precision than recall.

B. Sampling the Model

Once the model has been selected, and the parameters estimated by solving the system of linear equations in Equation (5), we need to sample a new population of individuals. In the general case, the graph associated to a MN may contains cycles, so that is not possible to employ Probabilistic Logic Sampling such as in BOA [6]. In order to sample new candidate solutions, we employ an iterated Monte Carlo method, and in particular we use a Gibbs sampler. Such sampler, described in Algorithm 3, works by iteratively sample each bit from the conditional probability of X_i given its



(a) (b) (c)

Fig. 2. Evaluation of different model selection techniques applied for the recovery of the network associated to a 8×8 variables 2D spin glass, with ± 1 coefficients. Red: ℓ_1-regularized logistic regression, truncation selection; green: CE, $sig = 1.50$; blue: χ^2, $q = 3.84$ (95% significance level); purple: JEMP, $sig = 1.50$. Experiments average over 30 executions. X axis is the control parameter β and population size after selection. Curves can be compared directly, since all selection percentage are equal to $ps = 10\%$.

neighbourhood $X_{N(i)}$, (the set of variables in $\mathcal{M}$ that interact with X_i), that is defined as

$$p_i(x_i|x_{N(i)};T) = \frac{1}{1+\mathrm{e}^{(U(x^+)-U(x^-))/T}}, \tag{9}$$

where x^+ and x^- denote a point x where the variable x_i is set respectively to $+1$ and to -1, while T is a parameter known as temperature of the sampler.

Algorithm 3: GIBBS SAMPLER

```
Randomly choose an initial point x = (x_1, x_2, ..., x_n)
Set r = 0
repeat
    Set x^tmp = x
    for i ← 1 to n do
        r = r + 1
        T = 1/cr
        Set x_i = 1 with probability p(x_i|N(i);T) as in Eq. (9)
until x^tmp = x ;
return x
```

The sampler generates an initial random point x, then it samples the solution until an entire scan[3] over the individual does not produce any bit mutation, i.e., does not flip any bit. The temperature T is decreased according to the cooling schema $1/cr$, where c is the cooling rate, a constant specified by the user, and r is a counter of the current amount of bit sampled. In doing so, the sampler concentrates the probability mass over those configuration x that have high probability.

When the sampler stops its execution, the sampled solution x is evaluated, and, if it is optimal or good enough with respect to a given criterium, the procedure ends, otherwise, a new solution is sampled until a maximum number R of sampled individuals is reached. It is worth noticing that, when the estimated model is good enough, repeatedly sampling the model with different random starts will yield an optimum with high probability, so that, in the general case, DEUM$_{\ell_1}$ can be run for only one generation.

[3]A scan of a sampler over a solution x is defined as sampling once each variable x_i given an arbitrary order.

V. EXPERIMENTAL RESULTS

In this section we present an evaluation of the performance of DEUM$_{\ell_1}$, compared with other popular EDAs, when applied to the resolution of some benchmarks with increasing complexity: One Max, Alternated Bits and Ising Spin Glass.

We provide results concerning the average number of fitness evaluations and the average execution time required by the algorithms to find the optimal solution of the given problem. Reliable estimations of the performance are achieved by averaging the results over 30 independent executions of the algorithms over the same problem instance. The performance of the algorithms are plotted only when they reach a success rate of 1, i.e., when they found the optimal solution in all runs.

The performance of DEUM$_{\ell_1}$ are compared, depending on the benchmark, with those of the univariate algorithms PBIL [23] and DEUM$_d$ [5], the bivariate DEUM$_{chain}$ [24] and IsDEUM [18], which have fixed structure for the MNs, and with EDAs which implement some model selection technique, such as sBOA [6] and DEUM-CE [8]. The algorithms belonging to the DEUM framework are characterized by the following parameters: population size p, percentage selection ps, and cooling rate $c = 0.0005$ fixed for each test. In addition, DEUM-CE requires to know the maximum neighbourhood size mn, and the significance parameter sig that is equal to 1.5 according to [8]. PBIL algorithm is defined by the population size p, the percentage selection ps and the learning rate γ, while sBOA requires, besides the population size p and the selection pressure ps, also the percentage of elitism pe, and the maximum number of incoming edges mi.

The results presented here come from a tuning stage over the parameters of each algorithm for every benchmark, and refers to the optimal performance, even if the tuning may not have been exhaustive. The experiments were generated using the Evoptool toolkit [25], an open source platform for comparison of Evolutionary Algorithms. We included in this software package an implementation of the DEUM$_{\ell_1}$ algorithm.

A. The One Max Function

The ability to learn the conditional independence structure encoded in a graphical model should not preclude the solution

of problems of bounded difficulty, in particular when the problem may be represented by an independence model. The One Max function is one of such problems. Given a solution x, this function is defined as the count of bits x_i taking value 1, so that the optimal solution is the string of all $1s$.

In Figures 3(a) and 4(a), we present the results of a set of experiments, respectively for size $n = 64$ and $n = 100$, in which we compare the performance of DEUM$_{\ell_1}$ with those of other popular EDAs when applied to the One Max function. DEUM-CE and sBOA were set to the most conservative configuration that allows structure learning, i.e, $mi = 1$ and $mn = 1$. Our proposal algorithm performs similarly, sometimes better, that the other EDAs in terms of number of fitness evaluations. In Figure 4(a), where the size of the problem is $n = 100$, the DEUM$_{\ell_1}$ algorithm requires an execution time higher than its opponents.

B. The Alternated Bits Function

The Alternated Bits function introduces bivariate interactions among the variables of the problem. The benchmark is defined as the sum of adjacent bits taking opposite value, i.e., $f(x) = \sum_{i=1}^{n-1} |x_i - x_{i+1}|$. The interactions structure of Alternated Bits can be modelled through a chain, either directed or undirected.

In Figures 3(b) and 4(b), we summarize the performance of DEUM$_{\ell_1}$, together with those of other EDAs, when applied to the Alternated Bits function, respectively for the problem size of $n = 64$ and $n = 100$. The univariate PBIL algorithm has not been able to reach a success rate of the 100%, so that it does not appear in the results of this tests. DEUM$_{\ell_1}$ performs similarly to DEUM-CE in terms of fitness calls. It is worth noting that, DEUM-CE requires prior knowledge about the size of the neighbourhood of the nodes in the MN, so that, differently from our proposal. Compared with sBOA, DEUM$_{\ell_1}$ and DEUM-CE perform significantly better in terms of fitness calls. This is mainly due to the fact that such algorithms, differently from sBOA, are able to find the optimal solution in a unique generation. However, the use of logistic regression begins to show the need of a significant effort in terms of execution time required to find the optimal solution.

C. The 2D Ising Spin Glass Function

Finally, we evaluate the performance of DEUM$_{\ell_1}$ over the well known Ising Spin Glass [2]. This function has a structure of interactions among variables that can be represented as a 2D lattice. This structure is hard to be efficiently modelled through a DAG, consequently, the sBOA algorithm is not able to find the global optimum on 50% of the tests[4]. Similarly, the independence model is not suitable for this benchmark, and PBIL is not able to converge to the optimum.

In Figures 3(c) and 4(c), we provide the results of the performance evaluation of the algorithms IsDEUM, DEUM-CE and DEUM$_{\ell_1}$ for size of $n = 64$ and $n = 100$.

[4]The hierarchical version of BOA, hBOA, has been successfully applied to the resolution of the Ising Spin Glass problem, see [26].

In order to find the optimal solution, DEUM$_{\ell_1}$ requires an average number of fitness evaluations similar to those required by the DEUM-CE. However, the amount of time spent in learning the structure of the interaction by DEUM$_{\ell_1}$ is significantly higher with respect to that required by DEUM-CE.[5] It is worth noting that, the higher execution time is balanced by the absence of prior knowledge required by the algorithm concerning the optimization problem. Indeed, if we underestimate the size of the neighbourhood in DEUM-CE, i.e., $m < 4$, our experiments show that this algorithm is not able to find the ground states of the Ising Spin Glass problem. We do not provide here the results of such experiments for reasons of space. We refer to the best performance of the algorithms after an experimental tuning of the parameters.

VI. Conclusions

In this paper we presented DEUM$_{\ell_1}$, a Markov Network based EDA able to perform consistent model selection without the use of prior knowledge concerning the objective function. DEUM$_{\ell_1}$ introduces the use of ℓ_1-regularized logistic regression, a powerful mean to deal with sparse models, in the context of EDAs based on undirected graphical models. We provided guidelines on the dimensioning of the strength of the regularizer parameter λ, and on the minimum size of the population to reach consistent model selection. We showed that our proposed approach to model selection is able to reconstruct correctly the interactions present in the fitness function, and from this point of view it behaves similarly, sometimes better, than other popular EDAs. In particular we remark that ℓ_1-regularized logistic regression does not require any additional information about the topology of the graphical model employed, even if, from preliminary experimental results, at the cost of a greater computational effort.

References

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

[2] R. Kindermann and J. Snell, *Markov random fields and their applications*. AMS, 1980.

[3] H. Mühlenbein, T. Mahnig, and A. Rodriguez, "Schemata, distributions and graphical models in evolutionary optimization," *Journal of Heuristics*, vol. 5, no. 2, pp. 215–247, 1999.

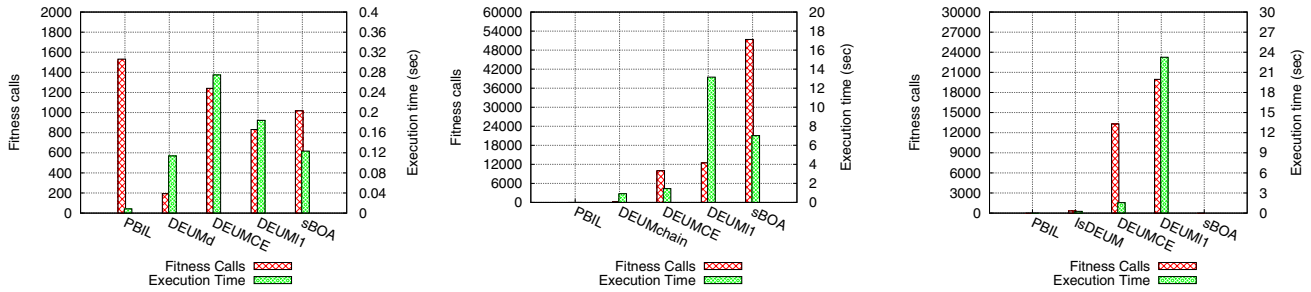
[4] R. Santana, "Estimation of Distribution Algorithms with Kikuchi Approximations," *Evolutionary Computation*, vol. 13, no. 1, pp. 67–97, 2005.

[5] S. Shakya and J. McCall, "Optimization by Estimation of Distribution with DEUM framework based on Markov random fields," *International Journal of Automation and Computing*, vol. 4, no. 3, pp. 262–272, 2007.

[6] M. Pelikan, D. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian Optimization Algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, vol. 1. Morgan Kaufmann Publishers, 1999, pp. 525–532.

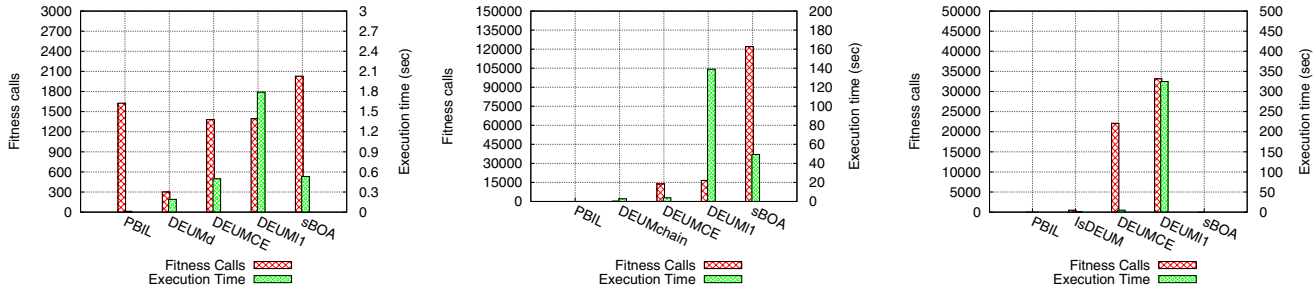
[7] M. Pelikan and D. Goldberg, "Escaping hierarchical traps with competent genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-01*. Morgan Kaufmann Publishers, 2001, pp. 511–518.

[5]In black-box contexts the evaluation of the fitness function is usually assumed to be the computational bottle neck. Since during the evaluation of the performance of the algorithm we deal with synthetic benchmarks whose computational cost is minimal, it is common practice to pay more attention to the amount of fitness calls then the effective execution time.



(a) (b) (c)

Fig. 3. Problem size $n = 64$. (a) One Max: PBIL $p = 128$, $ps = 50\%$, $\gamma = 1.0$; DEUM$_d$ $p = 192$, $ps = 50\%$; DEUM-CE $p = 1240$, $ps = 10\%$, $mn = 1$; DEUM$_{\ell_1}$ $p = 830$, $ps = 10\%$; sBOA $p = 128$, $ps = 50\%$, $pe = 50\%$, $mi = 1$; (b) Alt. Bits: PBIL success rate < 1; DEUM$_{chain}$ $p = 192$, $ps = 50\%$; DEUM-CE $p = 9980$, $ps = 10\%$, $mn = 2$; DEUM$_{\ell_1}$ $p = 12470$, $ps = 10\%$; sBOA $p = 6400$, $ps = 50\%$, $pe = 25\%$, $mi = 3$; (c) Ising Spin Glass: PBIL, sBOA success rate < 1; IsDEUM $p = 320$, $ps = 50\%$; DEUM-CE $p = 13300$, $ps = 10\%$, $mn = 4$; DEUM$_{\ell_1}$ $p = 19960$, $ps = 10\%$.



(a) (b) (c)

Fig. 4. Problem size $n = 100$. (a) One Max: PBIL $p = 100$, $ps = 50\%$, $\gamma = 1.0$; DEUM$_d$ $p = 300$, $ps = 50\%$; DEUM-CE $p = 1380$, $ps = 10\%$, $mn = 1$; DEUM$_{\ell_1}$ $p = 1380$, $ps = 10\%$; sBOA $p = 200$, $ps = 50\%$, $pe = 50\%$, $mi = 1$; (b) Alt. Bits: PBIL success rate < 1; DEUM$_{chain}$ $p = 300$, $ps = 50\%$; DEUM-CE $p = 13810$, $ps = 10\%$, $mn = 2$; DEUM$_{\ell_1}$ $p = 16570$, $ps = 10\%$; sBOA $p = 10000$, $ps = 50\%$, $pe = 25\%$, $mi = 4$; (c) Ising Spin Glass: PBIL. sBOA success rate < 1; IsDEUM $p = 500$, $ps = 50\%$; DEUM-CE $p = 22100$, $ps = 10\%$, $mn = 4$; DEUM$_{\ell_1}$ $p = 33150$, $ps = 10\%$.

[8] A. Brownlee, J. McCall, S. Shakya, and Q. Zhang, "Structure learning and optimisation in a Markov Network based Estimation of Distribution Algorithm," *Exploitation of Linkage Learning in Evolutionary Algorithms*, pp. 45–69, 2010.

[9] S. Shakya, A. Brownlee, J. McCall, F. Fournier, and G. Owusu, "A fully multivariate DEUM algorithm," in *IEEE Congress on Evolutionary Computation, 2009.* IEEE, 2009, pp. 479–486.

[10] L. Malagò, M. Matteucci, and G. Pistone, "Towards the geometry of Estimation of Distribution Algorithms based on the exponential family," in *Proceedings of XI Foundation of Genetic Algorithms (FOGA)*, 2011.

[11] Q. Zhang, J. Sun, and E. Tsang, "An evolutionary algorithm with guided mutation for the maximum clique problem," *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 2, pp. 192 – 200, 2005.

[12] J. Yang, H. Xu, Y. Cai, and P. Jia, "Effective structure learning for EDA via L1-regularized bayesian networks," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation, GECCO-01*. ACM, 2010, pp. 327–334.

[13] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Regularized model learning in estimation of distribution algorithms for continuous optimization problems," UPM-FI/DIA/2011-1, Tech. Rep., 2011.

[14] L. D. Brown, *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*, ser. Lecture Notes - Monograph Series. California: Institute of Mathematical Statistics, 1986, vol. 9.

[15] E. Boros and P. L. Hammer, "Pseudo-boolean optimization," *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 155–225, 2002.

[16] H. Mühlenbein and T. Mahnig, "Mathematical analysis of evolutionary algorithms," in *Essays and Surveys in Metaheuristics*. Kluwer Academic Publisher, 2002, pp. 525–556.

[17] D. Brown, A. Garmendia-Doval, and J. McCall, "Markov random field modelling of royal road genetic algorithms," in *Artificial Evolution*. Springer, 2002, pp. 35–56.

[18] S. Shakya, J. McCall, and D. Brown, "Solving the Ising spin glass problem using a bivariate EDA based on Markov random fields," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, 2006, pp. 908–915.

[19] R. Heckendorn and A. Wright, "Efficient linkage discovery by limited probing," *Evolutionary computation*, vol. 12, no. 4, pp. 517–545, 2004.

[20] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty, "High-dimensional Ising model selection using ℓ_1-regularized logistic regression," *The Annals of Statistics*, vol. 38, no. 3, pp. 1287–1319, 2010.

[21] K. Koh, S. Kim, and S. Boyd, "An interior-point method for large-scale ℓ_1-regularized logistic regression," *Journal of Machine Learning Research*, vol. 8, no. 8, pp. 1519–1555, 2007.

[22] M. De la Maza and B. Tidor, "An analysis of selection procedures with particular attention paid to proportional and Boltzmann selection," in *Proceedings of the 5th International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, 1993, pp. 124–131.

[23] S. Baluja and R. Caruana, "Removing the genetics from the standard genetic algorithm," in *Machine learning: proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 38–46.

[24] A. E. I. Brownlee, "Multivariate Markov Networks for fitness modelling in an Estimation of Distribution Algorithm," Ph.D. dissertation, The Robert Gordon University, 2009.

[25] G. Valentini, L. Malagò, and M. Matteucci, "Evoptool: an extensible toolkit for evolutionary optimization algorithms comparison," in *Proceedings of IEEE World Congress on Computational Intelligence*, July 2010, pp. 2475–2482.

[26] M. Pelikan, K. Helmut, and S. Kobe, "Finding ground states of Sherrington-Kirkpatrick spin glasses with hierarchical boa and genetic algorithms," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, ser. GECCO '08. New York, NY, USA: ACM, 2008, pp. 447–454.