Toward Understanding EDAs Based on Bayesian Networks Through a Quantitative Analysis

Carlos Echegoyen, Alexander Mendiburu, Roberto Santana, and Jose A. Lozano, *Member, IEEE*

***Abstract*—The successful application of estimation of distribution algorithms (EDAs) to solve different kinds of problems has reinforced their candidature as promising black-box optimization tools. However, their internal behavior is still not completely understood and therefore it is necessary to work in this direction in order to advance their development. This paper presents a methodology of analysis which provides new information about the behavior of EDAs by quantitatively analyzing the probabilistic models learned during the search. We particularly focus on calculating the probabilities of the optimal solutions, the most probable solution given by the model and the best individual of the population at each step of the algorithm. We carry out the analysis by optimizing functions of different nature such as Trap5, two variants of Ising spin glass and Max-SAT. By using different structures in the probabilistic models, we also analyze the impact of the structural model accuracy in the quantitative behavior of EDAs. In addition, the objective function values of our analyzed key solutions are contrasted with their probability values in order to study the connection between function and probabilistic models. The results not only show information about the internal behavior of EDAs, but also about the quality of the optimization process and setup of the parameters, the relationship between the probabilistic model and the fitness function, and even about the problem itself. Furthermore, the results allow us to discover common patterns of behavior in EDAs and propose new ideas in the development of this type of algorithms.**

***Index Terms*—Abductive inference, Bayesian networks, estimation of Bayesian networks algorithm, estimation of distribution algorithms, Ising, Max-SAT, probabilistic model.**

I. Introduction

ESTIMATION of distribution algorithms (EDAs) [1]–[3] are a population-based optimization paradigm in the field of evolutionary computation that have acquired special relevance in the last decade. Nowadays, they are a strong alternative for solving optimization problems from different domains such as engineering [4], biomedical informatics [5], [6], or robotics [7]. However, despite their successful application there are a wide variety of open questions [8] regarding the behavior of this type of algorithms.

The main characteristic of EDAs is the use of probabilistic models to lead the search toward promising areas of the space of solutions. By making use of a subset of promising solutions belonging to the population, the employed probabilistic models allow us to estimate a new probability distribution over the search space at each step of the algorithm. Thus, each of the possible solutions has an associated probability of being sampled which varies during the optimization process. The probability values assigned to the solutions are the main source in determining which solutions will be returned by the algorithm. Consequently, given a problem, the fundamental objective is to get higher probability values for the highest quality solutions throughout an iterative process. Naturally, to reach the optimal solution is an inherent challenge and a reference in the development of both theoretical [9] and practical [10] EDAs.

In order to better understand how these algorithms solve the problems, the characteristics of the probabilistic models used are a rich source of information which has been studied in several works [11]–[17]. In particular, one class of model that has been extensively applied in EDAs are Bayesian networks [18], which allow to encode probability distributions through a structure, that expresses explicit independence relations among variables, and a set of parameters. There are different implementations of EDAs based on this type of models [19]–[21]. A straightforward form of analysis when Bayesian networks are used is through the explicit interactions among the variables they provide. Thus, it has been shown how different parameters of the algorithm influence the accuracy of the structural models [16], how the dependences of the probabilistic models change during the search [15] and, moreover, how the networks learned can provide information about the problem structure [14], [15], [22].

With the aim of continuing the study of EDAs, we take a different but complementary path which was initiated in a preliminary version [23] of this paper. Specifically, we propose a new methodology based on a quantitative analysis of the probabilistic models. As we have argued, the particular probability values assigned to the solutions during the search are the raw material from which EDAs obtain the results. Therefore, studying such probabilities provides useful new information

Manuscript received September 21, 2009; revised March 30, 2010, September 2, 2010, and November 2, 2010; accepted December 4, 2010. Date of publication January 28, 2011; date of current version March 30, 2012. This work was supported in part by the Saiotek and Research Groups 2007–2012 (IT-242-07) Programs (Basque Government), TIN2008-06815-C02-01, TIN2010-14931, and Consolider Ingenio 2010-CSD2007-00018 Projects (Spanish Ministry of Science and Innovation) and COMBIOMED network in computational biomedicine (Carlos III Health Institute). C. Echegoyen holds a grant from UPV-EHU. The work of R. Santana was supported in part by the TIN2010-20900-C04-04 and Caja Blue Brain Project (Spanish Ministry of Science and Innovation).

C. Echegoyen, A. Mendiburu, and J. A. Lozano are with the Intelligent Systems Group, University of the Basque Country, San Sebastián-Donostia 20018, Spain (e-mail: carlos.echegoyen@ehu.es; alexander.mendiburu@ehu.es; ja.lozano@ehu.es).

R. Santana is with the Cajal Blue Brain Project, Technical University of Madrid, Madrid 28660, Spain (e-mail: roberto.santana@upm.es).

Digital Object Identifier 10.1109/TEVC.2010.2102037

to better understand the behavior of this type of algorithm. Following this criterion, our quantitative analysis of EDAs is based on monitoring the probability of certain distinguished solutions during the search: 1) the optimal solution of the function; 2) the solution with the highest probability in the distribution; and 3) the best individual in each generation. In order to complete the quantitative analysis, we also record the fitness function values for solutions 2) and 3) during the search.

In particular, the proposed analysis is carried out when the estimation of Bayesian networks algorithm (EBNA) [19] is applied to problems of different nature. We use different structural models which can be learned from the population or created by reproducing interactions among the variables of the problem. We also use different population sizes in order to analyze the influence of this parameter in the algorithm. Furthermore, we take into account both successful (the optimum is reached) and unsuccessful runs (the optimum is not reached).

Throughout this paper we shed light on basic questions of great interest that still remain open in EDAs such as the following.

1) How does the probability assigned by the probability distributions to the optimal solution evolve during the search?

This first question plays a key role in this paper and it is related with a number of current assumptions in the application of EDAs. For example, whether, in order to solve a problem, it is a necessary condition that the probability associated by the algorithm to the optimal solution steadily increases at each generation or whether the highest probability is assigned to the optimum during the search.

2) How does the accuracy of the information about the problem contained in the structural model influence the internal behavior of EDAs?

This question is addressed in order to better know the relation between the interactions of the problem and the dependences of the probabilistic model. By using different structures in EDAs, we study the effect that introducing more interactions in the structural model has on the behavior of EDAs in general and in the previously mentioned assumptions about the probability of the optimum in particular.

Previous works [15], [24] have considered different means of introducing *a priori* knowledge of the problem into the algorithm in order to improve the efficiency and efficacy of EDAs. Understanding the impact of this type of practices in the internal behavior of EDAs is also a very important issue in their application to real problems.

3) How does the function value for the most probable solution given by the probabilistic models evolve during the search?

A contribution of this paper is the exact calculation and analysis of the solution with the highest probability in the distributions estimated at each generation. Thus, by using its fitness function value, we can study how the probabilistic model captures the properties of the function. It would be desirable that the function value of the solution with the highest probability increased during the search.

Although the main target of this paper is to provide insights about these key issues, the results obtained show a different perspective of EDAs that is able to reveal constant patterns in their behavior. Furthermore, both the probability and function values analyzed are able to capture the quality of the probabilistic models in terms of their use within EDAs. Throughout the analysis proposed, it is also possible to better understand how the convergence of the algorithm occurs and even detect multimodality in the problems solved.

Finally, based on the conclusions of the work, we are able to propose different ideas in order to assist in the use of EDAs in real problems and contribute to their development. The main proposals are related to: 1) bringing forward premature convergence by detecting crucial phases of the search; 2) measuring the influence of different components of the algorithm and their impact in the search when the optimum is unknown; 3) on-line monitoring of the optimization process allowing certain automatic decision making; and 4) taking advantage of the available information of the problem.

The rest of this paper is organized as follows. Section II presents EBNA, and introduces Bayesian networks and abductive inference. Section III explains the experimental design. Sections IV, V, VI, and VII discuss Trap5, Gaussian Ising, $\pm J$ Ising, and Max-SAT problems, respectively, quantitatively analyzing the behavior of EDAs for each problem when different structural models and population sizes are used. Section VIII discusses relevant previous works. Section IX draws the conclusion obtained during the study. Finally, Section X points out possible future studies.

II. Background

A. Notation

Let X be a random variable, a value of X is denoted by x. $\boldsymbol{X} = (X_1, \ldots, X_n)$ will denote a vector of random variables. We will use $\boldsymbol{x} = (x_1, \ldots, x_n)$ to denote an assignment to the variables. Each variable X_i has r_i possible values, $x_i^1, \ldots, x_i^{r_i}$. We will work with discrete variables. The joint probability distribution of $\boldsymbol{X}$ is represented as $p(\boldsymbol{X} = \boldsymbol{x})$ or $p(\boldsymbol{x})$. We use $p(X_i = x_i | X_j = x_j)$ or, in a simplified form, $p(x_i|x_j)$, to denote the conditional probability distribution of X_i given $X_j = x_j$.

B. Bayesian Networks

Formally, a Bayesian network is a pair $(S, \boldsymbol{\theta})$ representing a graphical factorization of a probability distribution. The structure S is a directed acyclic graph which reflects the set of conditional (in)dependences among the variables. On the other hand, $\boldsymbol{\theta}$ is a set of parameters for the local probability distributions associated with each variable.

The factorization of the probability distribution is codified as

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i | \boldsymbol{pa}_i) \tag{1}$$

where $\boldsymbol{pa}_i$ denotes a value of the variables $\boldsymbol{Pa}_i$, the parent set of X_i in the graph S.

Algorithm 1 EBNA

1: $BN_0 \leftarrow (S_0, \boldsymbol{\theta}^0)$ where S_0 is an arc-less structure, and θ^0 is uniform
2: $D_0 \leftarrow$ Sample N individuals from BN_0
3: t $\leftarrow 1$
4: **do** $\{$
5: $D_{t-1} \leftarrow$ Evaluate individuals
6: $D_{t-1}^{Se} \leftarrow$ Select $N/2$ individuals from D_{t-1}
7: $S_t^* \leftarrow$ Obtain a network structure
8: $\boldsymbol{\theta}^t \leftarrow$ Calculate θ_{ijk}^t using D_{t-1}^{Se} as the data set
9: $BN_t \leftarrow (S_t^*, \boldsymbol{\theta}^t)$
10: $D_t \leftarrow$ Sample $N-1$ individuals from BN_t and create the new population
11: $\}$**until** Stop criterion is met

With reference to the set of parameters $\boldsymbol{\theta}$, if the variable X_i has r_i possible values, the local distribution $p(x_i|\boldsymbol{pa}_i^j, \boldsymbol{\theta}_i)$ is an unrestricted discrete distribution

$$p(x_i^k|\boldsymbol{pa}_i^j, \boldsymbol{\theta}_i) \equiv \theta_{ijk} \tag{2}$$

where $\boldsymbol{pa}_i^1, \ldots, \boldsymbol{pa}_i^{q_i}$ denote the q_i possible values of the parent set $\boldsymbol{Pa}_i$. In other words, the parameter θ_{ijk} represents the probability of variable X_i being in its kth value, knowing that the set of its parents' variables is in its jth value. Therefore, the local parameters are given by $\boldsymbol{\theta}_i = ((\theta_{ijk})_{k=1}^{r_i})_{j=1}^{q_i}$. The introduction of Bayesian networks in EDAs requires appropriate methods of learning and sampling. As has been shown, to complete the network learning it is necessary to obtain the structure S and the set of parameters $\boldsymbol{\theta}$.

When the structure is not given, it can be learned from a data set. We use a structural learning algorithm based on a "score+search" technique [2]. Specifically, the search is carried out using B algorithm [25] and the score is the Bayesian information criterion (BIC) [26]. Regarding the second step, the parameters $\boldsymbol{\theta}$ of the Bayesian network are estimated by maximum likelihood using Laplace correction [2]. Finally, to sample the Bayesian network, a forward sampling method is used. A variable is sampled once all its parents have been sampled. This method is known as probabilistic logic sampling (PLS) [27].

C. EDAs Based on Bayesian Networks

Following the main scheme of EDAs, EBNA [19] works with populations of N individuals that constitute sets of N candidate solutions. The initial population is generated according to a uniform distribution, and hence, all the solutions have the same probability of being sampled. Each iteration starts by selecting a subset of promising individuals from the population. Although there are different selection methods, in this case we make use of truncation selection with threshold 50%. Thus, the $N/2$ individuals with the best fitness value are selected. The next step is to learn a Bayesian network from the subset of selected individuals. Once the Bayesian network is built, the new population can be generated. At this point there are different possibilities. We use elitism because it is a classic technique widely used in evolutionary computation. From the Bayesian network, $N-1$ new solutions are sampled and then added to the N individuals of the current population. The N best individuals, among the $2N-1$ available, constitute the new population. By using an elitist criterion, once an optimal solution is reached by the algorithm, the solution will be kept in the population until the end of the run. Thus, we can monitor and analyze the same optimum throughout the generations.

The procedure of selection, learning, and sampling is repeated until a stop condition is fulfilled. A pseudocode of EBNA is shown in Algorithm 1.

D. Abductive Inference in Bayesian Networks

In general, abductive reasoning tries to find the hypothesis that would best explain a set of facts or observations. In the probabilistic network context, the abductive inference [28] consists of finding the maximum *a posteriori* probability state of the network variables, given some evidence (observed variables).

The total abductive inference involves all the problem variables and is defined as follows. Given a probability distribution over the vector of random variables $\boldsymbol{X}$ and the evidence $\boldsymbol{e}$, that is an instance of the observed variable set $\boldsymbol{E} \subseteq \boldsymbol{X}$, we want to obtain the assignment $\boldsymbol{x}_{\boldsymbol{U}}^*$ to the unobserved variables $\boldsymbol{X}_{\boldsymbol{U}} = \boldsymbol{X} \setminus \boldsymbol{E}$ such that

$$\boldsymbol{x}_{\boldsymbol{U}}^* = \arg\max_{\boldsymbol{x}_{\boldsymbol{U}}} p(\boldsymbol{x}_{\boldsymbol{U}}|\boldsymbol{e}). \tag{3}$$

Usually, $\boldsymbol{x}_{\boldsymbol{U}}^*$ is known as the *most probable explanation*.

However, when this technique is applied to the probability distributions associated with Bayesian networks in EDAs, there is no evidence. In this case, the objective is to look for the assignment $\boldsymbol{x}^*$ with the highest probability for the whole set of variables $\boldsymbol{X}$. Knowing that $P(\boldsymbol{X}_{\boldsymbol{U}}|\boldsymbol{E}) = P(\boldsymbol{X}|\boldsymbol{E})$ and having an empty evidence set $\boldsymbol{E} = \emptyset$, (3) can be directly converted into our target

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\hat{\boldsymbol{\theta}}, \hat{S}) \tag{4}$$

where $\hat{S}$ is the structure of the model which has been learned from the population by using the BIC score and $\hat{\boldsymbol{\theta}}$ represents the parameters of the probabilistic model estimated by maximum likelihood. In our context of EDAs, $\boldsymbol{x}^*$ is called the most probable solution (MPS). As is proven in [29], this kind of inference is an NP-hard problem. Therefore, its exact resolution is only feasible in problems of limited length.

In this paper, the point with the highest probability is calculated using probability propagation in junction trees [30] or variable elimination techniques [31], as they are implemented in Bayes Net Toolbox [32].

III. Experimental Design

The experiments were mainly designed with the aim of shedding light on the questions and assumptions mentioned in Section I. Specifically, at each generation of EBNA, we record the probability and fitness values of distinguished solutions of the search space: the optimum, the most probable solution, and the best individual in the population. By varying the problem size, using different structural modes and different

population sizes, we create different scenarios to complete the analysis.

In order to show the relation between the probabilities of our distinguished solutions and the diversity of the population, we hereby introduce additional information that is not obtained from the probabilistic model but directly from the population itself. Particularly, at each step of the algorithm, we calculate the accumulated entropy of the population by means of adding the entropy of each variable belonging to the function

$$H(\boldsymbol{X}) = -\sum_{i=1}^{n}\sum_{j=1}^{r_i} p(x_i^j)\cdot \log_2 p(x_i^j). \tag{5}$$

This metric shows how the population managed by the algorithm loses diversity and converges. Some works have already studied these types of measurements in EDAs [33], [34].

In the following section, we will explain the different problems, structural models, and parameters used for the experiments. In [35], the necessary tools to reproduce the experiments or to carry out similar analysis are implemented.

A. Problems

The whole set of problems is based on additively decomposable functions (ADFs) defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} f_i(\boldsymbol{s_i}) \tag{6}$$

where $\boldsymbol{S_i} \subseteq \boldsymbol{X}$. With the aim of covering a wide spectrum of applications and observing the behavior of EDAs in different scenarios, we chose the following four test problems: Trap5, Gaussian Ising, $\pm J$ Ising, and Max-SAT. The exact details of each problem will be introduced in the following sections. These problems are selected for several reasons. First, in order to investigate the influence of multimodality in the behavior of EDAs, we deal with problems that have different numbers of optimal solutions. The first two problems have just one optimal solution and the last two problems have several optimal solutions. Second, all of them are optimization problems which have been widely used to analyze EDAs [15], [36], [37]. Finally, all the problems have a different nature. Trap5 [38] is a deceptive function designed in the context of genetic algorithms [39] aimed at finding their limitations. It is a separable function and in practice it can be easily optimized if the structure is known. Gaussian Ising and $\pm J$ Ising come from statistical physics domains and are instances of the Ising model proposed to analyze ferromagnetism [40]. The variables are disposed on a grid and the interactions do not allow us to decompose the problem into independent subproblems of bounded order [41]. It is a challenge in optimization [15], [36] and in its general form is NP-complete [42]. Max-SAT is a variation for optimization of a classic benchmark problem in computational complexity, the propositional satisfiability or SAT. In fact, SAT was the first problem proven to be NP-complete [43] in its general form. An instance of this problem can contain a very high number of interactions among variables and in general, it cannot be efficiently divided into subproblems of bounded size in order to reach the optimum. With the exception of the function Trap5, we have dealt with 100 instances for each type of problem.

Regarding the information stored at each generation, when the functions with just one optimum are optimized, we only record our three distinguished solutions. However, in the functions with several optimal solutions, EBNA reaches a subset of those optima and the analysis of the probability of the optimum is extended. Thus, we calculate the probabilities during the search for all optima reached by EBNA in the last generation. It leads us to see how the probability is distributed when there are different optimal solutions. In order to gain clarity in the results, we only show the maximum and minimum probabilities assigned by the probability distribution to the reached optimal solutions at each generation.

B. Structural Models

In the literature, several works have discussed the influence of the structure of the probabilistic model in the behavior of EDAs [14], [22] and the impact of using *a priori* knowledge of the problem in the search [15], [24]. In this paper, we also take into account these important issues. Therefore, in addition to traditional learning techniques, we propose to include some structures related with the problem to analyze the changes produced in the internal behavior of EDAs.

Specifically, we use the following two approaches as regards the structural models. On the one hand, we use an approximate structural learning algorithm (B algorithm) to obtain a new structure from the selected individuals at each generation. On the other, we use two fixed structures related to the nature of the function, and thus, only parametric learning is carried out. Since all the functions are ADFs, an intuitive and straightforward way to create a related structural model is by means of linking variables belonging to the same subfunction with arcs. In order to analyze the influence of the information accuracy introduced in the structural model, two structures have been used. The first structure tries to reproduce all interactions among variables that can be directly observed from the formulation of the problem. As a general method, we connect two variables (representing nodes in the graph) by an edge in the structure if the corresponding variables are contained in the same sub-function. Then, by taking a directed acyclic version of this undirected graph, we obtain a Bayesian network structure which will be called *dense structure*. The second structure also reproduces interactions among the variables of the function but only considers bivariate dependences. This structure has less information but is always related with the nature of the problem. It will be called *bivariate structure*.

It must be pointed out that our aim is to study the influence that the different approaches have over the probability values rather than demonstrating their quality and accuracy.

C. Parameter Configuration

The sample size is very important in order to learn Bayesian networks [44] and, hence, in the behavior of EDAs based on this type of models. Thus, we deal with two different population sizes in order to analyze their influence in the algorithm. First, we use the bisection method [3] to determine an adequate population size to reach the optimum (with high

probability). This size is denoted by m. The stopping criterion for bisection is to obtain the optimum in 10 out of 10 independent runs. The final population size is the average over 20 successful bisection runs. Due to computational restrictions, the maximum population size has been limited to 2^{14}. The population size m is always obtained from EBNA executions with B algorithm. The second population size is half of the bisection, $m/2$. With this size we try to create a more realistic scenario in which achieving the optimum is less likely. This also allows us to analyze in detail the probability of the optimum when it is not reached.

In addition to population size, different problem dimensions have also been taking into account. Particularly, we have used $n \in \{50, 75, 100\}$ for Trap5 and Max-SAT, and $n \in \{8 \times 8, 9 \times 9, 10 \times 10\}$ for both types of Ising. The upper bound has been set to 100 variables due to the high memory requirements needed to calculate the most probable solution. Increasing the number of variables would require the use of approximate inference techniques [45], spoiling the correctness of the results.

The stopping criterion for EBNA is a fixed number of iterations and is independent of obtaining the optimum. Each execution will run n generations, that is, as many as the number of problem variables. This number of generations is enough to observe the convergence of the analyzed probability values.

D. Details of the Experiments

The analyzed probability values are reported in logarithmic scale in order to smoothen the probability slopes and better observe their behavior from the beginning of the run. The number of runs which have reached the optimum at each generation is indicated with bars on the top of the charts, where the probability values are shown. Although we have made runs with a fixed number of generations, the charts presented were cut when all runs have reached the optimum or the curves are stabilized.

Concerning the total number of executions, for each Bayesian network learning approach and population size, we carried out 50 independent runs for Trap5 and 5 independent runs for each of the 100 instances in the rest of the problems. All the runs belonging to 100 different instances of a given problem are put together and analyzed as a whole in order to provide a general view of the behavior of EDAs.

Analyzing the wide set of results collected throughout the experiments, we have observed that EDAs show the same patterns of behavior independently of the problem dimension. Therefore, we will focus on problem sizes of 100 variables. For the sake of simplicity, in this paper we only present the most relevant results. However, for the interested reader, the complete analysis is available on the website of the Intelligent Systems Group.[1]

IV. EDA Behavior Solving Trap5

A. Trap5 Description

Our first function, Trap5 [38], is an additively separable (non overlapping) function with a unique optimum. It divides the set $\boldsymbol{X}$ of n variables into disjoint subsets $\boldsymbol{X}_I$ of 5 variables. It can be defined using a unitation function $u(\boldsymbol{y}) = \sum_{i=1}^{p} y_i$ where $\boldsymbol{y} \in \{0, 1\}^p$ as

$$\text{Trap5}(\boldsymbol{x}) = \sum_{I=1}^{\frac{n}{5}} trap_5(\boldsymbol{x}_I) \tag{7}$$

where $trap_5$ is defined as

$$trap_5(\boldsymbol{x}_I) = \begin{cases} 5, & \text{if } u(\boldsymbol{x}_I) = 5 \\ 4 - u(\boldsymbol{x}_I), & \text{otherwise} \end{cases} \tag{8}$$

and $\boldsymbol{x}_I = (x_{5I-4}, x_{5I-3}, x_{5I-2}, x_{5I-1}, x_{5I})$ is an assignment to each trap partition $\boldsymbol{X}_I$. This function has one global optimum in the assignment of all ones for $\boldsymbol{X}$ and a large number of local optima, $2^{n/5} - 1$.

Trap5 function has been used in previous works [15] to study the structure of the probabilistic models in EDAs based on Bayesian networks, as well as studying the influence of different parameters [16]. It is important to note that this function is difficult to optimize if the probabilistic model is not able to identify interactions between variables [23].

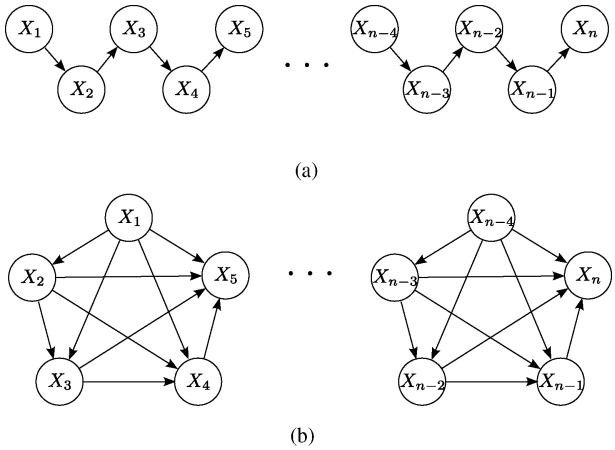

Fig. 1. Fixed structural models related to the dependences among the variables in Trap5. (a) Dense structure. (b) Bivariate structure.

B. Structures Related to the Problem

In this section, we propose two fixed structures related to the Trap5 function. The dense structure is created by linking all the variables in each sub-function $trap_5$. Thus, by providing direction to the arcs without creating cycles, we obtain the Bayesian network structure shown in Fig. 1(a). With this structure, there are no independences between variables of the same subgroup and variables in different partitions are independent. For this type of separable functions, this intuitive method to introduce information of the problem into the structural model, provides exact factorizations [41]. However, this is not the case for the rest of the problems. In general, the construction of exact factorizations could require additional techniques [46] and the resulting conditional probabilities would also require the estimation of a prohibitive number of parameters that would be unmanageable in practice.

As regards the bivariate structure, it is formed by a chain for each subgroup of 5 variables. As can be seen in Fig. 1(b),

[1]Available at http://www.sc.ehu.es/ccwbayes/members/carlos/edamps.

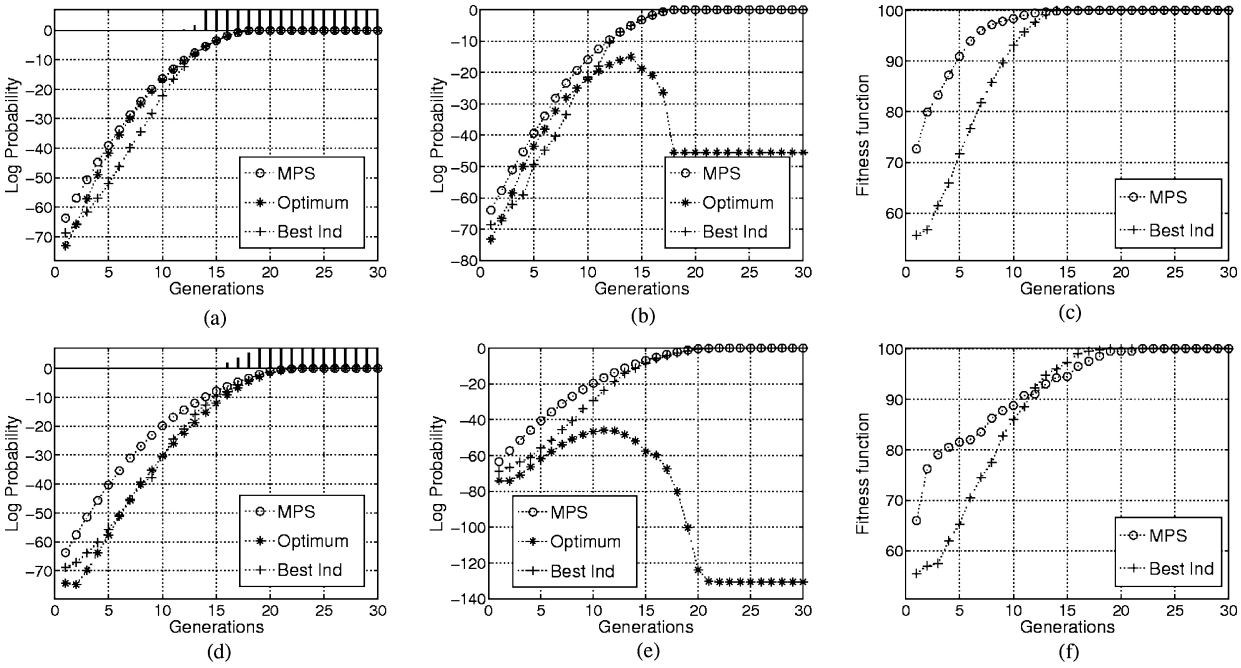

Fig. 2. Probability values and function values when EBNA is applied to Trap5 using Algorithm B. We have 49 out of 50 successful runs with population size m and 4 out of 50 with population size $m/2$. (a) Successful runs with population size m. (b) Unsuccessful runs with population size m. (c) Successful runs with population size m. (d) Successful runs with population size $m/2$. (e) Unsuccessful runs with population size $m/2$. (f) Successful runs with population size $m/2$.

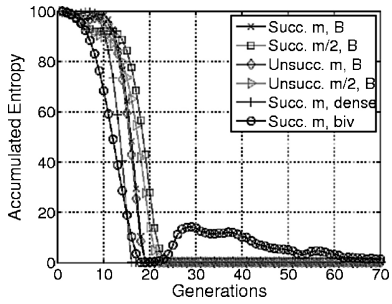

Fig. 3. Accumulated entropies of the population when EBNA is applied to Trap5.

the graph contains the minimum number of arcs necessary to connect all the variables belonging to each partition.

C. Using Structural Learning

In this section, we present and discuss the results obtained when EBNA, using Algorithm B, tries to optimize the Trap5 function.

In Fig. 2(a), we show the probability values for successful runs when the population size m (given by bisection) is used. In this case, EBNA reaches the optimum in 49 out of 50 runs. Theoretically, a convergence of the algorithm to the global optimum implies an increase in its probability value as the generations advance. This fact is reflected in the results. The probability values for the optimum and the MPS grow simultaneously, and very closely, when the executions are successful.

When a population size of $m/2$ [Fig. 2(d)] is used, the results change drastically and only four runs reach the optimum. Although the behavior of the probability values in this case is analogous to Fig. 2(a), we can observe that the probability curves for the MPS and the optimum with population size $m/2$ are clearly more distant than with size m. This analysis shows the impact that the population size has in the EDA behavior. Another important observation is that the growth of the probability values is slower for $m/2$. Thus, with this population size, the optimum is reached for the first time, a few generations later than with size m [bars on the top of Fig. 2(a) and (d)]. Nevertheless, with both population sizes, the executions starts to reach the optimum when its probability is approximately -10 in logarithmic scale.

The results of the probability values for the executions where the optimum was not reached are shown in Fig. 2(b) and (e). In these figures, we can see the joint growth of the probability values for the MPS and the optimum at the beginning of the run. However, after a certain generation, both values start to diverge and the optimum is no longer obtained. In unsuccessful runs, there are also differences between the runs with population size m and $m/2$. For this last population size, the probability of the optimum reaches lower values both before decreasing and in the last generations. Even at the beginning of the run, the probability of the optimum is further from the highest probability in the distribution with population size $m/2$ than with m.

Concerning the fitness function value, Fig. 2(c) shows the results when the population size m, given by bisection, is used in EBNA. The value of the MPS increases at each generation and it is better than the best individual in almost all generations. However, by looking at Fig. 2(f), it can bee seen that the MPS has a lower growth with population size $m/2$. Moreover, the best individual of the population is better than the MPS after generation 12. Therefore, the analysis of the function values also reflects the impact of the population size

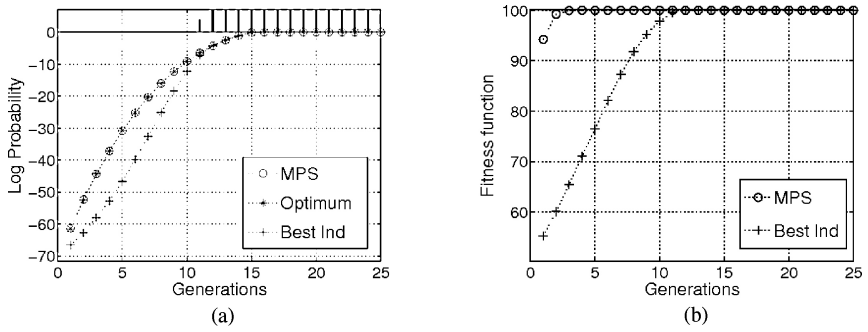

Fig. 4. Successful runs when EBNA is applied to Trap5 using the dense structure with population size m. The optimum is reached for the 50 executions.

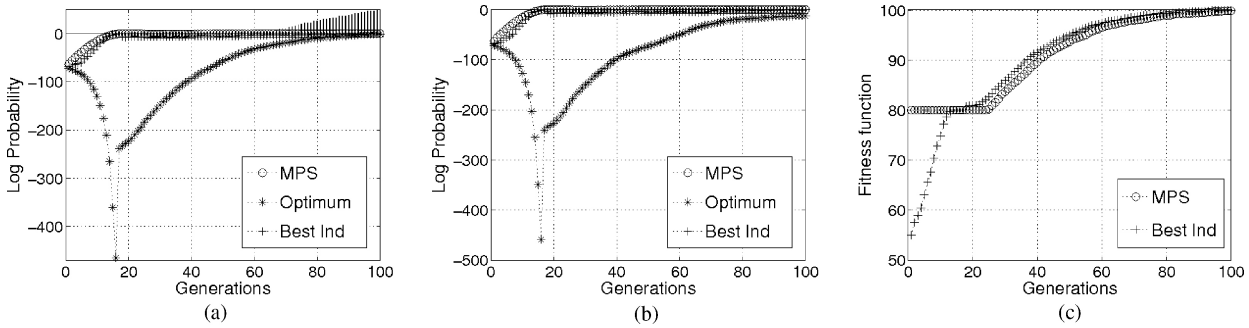

Fig. 5. Probability values and function values when EBNA is applied to Trap5 using the bivariate structure with population size m. We have 42 out of 50 successful executions. (a) Successful runs. (b) Unsuccessful runs. (c) Successful runs.

in the algorithm. In all the experiments, the curves of function values are similar in successful and unsuccessful runs.

Finally, in Fig. 3 we present the accumulated entropies of the population during the search. Only the first four curves shown in the legend correspond to EBNA with Algorithm B. These curves show the close relation of this measure with the exponential growth of the probability values. Moreover, we can observe how the population converges to a unique solution since the entropy tends to 0.

D. Using Fixed Structures

In this section, we show the behavior of the algorithm when a different amount of information is introduced in the structural model. As previously mentioned, throughout this paper we only show and comment on those results that provide relevant information, avoiding excessive and redundant information. So, when the structures are fixed, we have observed a low influence of the population size in the results. In these cases, we only show the analysis for the size m.

Fig. 4 shows the probability and fitness values when the dense structure is introduced in EBNA. In this case, we obtain an ideal behavior for an optimization process since the optimum has the highest probability during the whole run and is reached in all executions. Furthermore, in Fig. 4(b) we observe that the function values for MPS are close to the optimum from the very beginning of the search. In this case, through a sampling based on inference, the optimum could be reached in the first generations.

The behavior of the algorithm changes drastically when the bivariate structure [Fig. 1(b)] is introduced. Although EBNA shows a good performance because it reaches the optimum in 42 out of 50 runs, the evolution of the probability values [see Fig. 5(a)] in particular. The probability of the optimum decreases at the beginning of the run and when the algorithm seems to converge to a local optimum, it suddenly recovers. This fact also occurs for the unsuccessful executions [Fig. 5(b)] where the probability of the optimum increases in the last generations. In this case, the optimum has a high probability at the end of the run, which supports the belief that the algorithm would be able to reach the optimum provided that more generations are allowed.

The reason for such an uncommon behavior is the following. In the first part of the run, when the probability of the optimum decreases, the algorithm is deceived by the function and most of the individuals in the population become the local optimum. This local optimum is the assignment of zeros for all the set of variables $\boldsymbol{X}$ because it is the suboptimal value that $trap_5$ function gives to each trap partition $\boldsymbol{X}_I$. Fig. 3 shows how the curve of entropy for the bivariate model (Succ. m, biv) tends to 0 when the probability of the optimum is minimum in Fig. 5(a) and (b). After this stage, the algorithm recovers and the probability of the optimum begins to increase. The curve of entropy for the bivariate model indicates that different individuals are included in the population just when the algorithm seems to converge to the local optimum. It shows that the algorithm samples better individuals and is reflected in the fitness function values in Fig. 5(c). This can be explained through the Laplace correction and the fixed structure of chain subgraphs. This quantitative analysis justifies why it is possible to reach the optimum for this function with a simple bivariate structure.

While several works have analyzed EDA behavior using entropy measures [33], [34], [47], the joint analysis of the relationship between the type of model structure, the probability values and the entropy should support a more com-

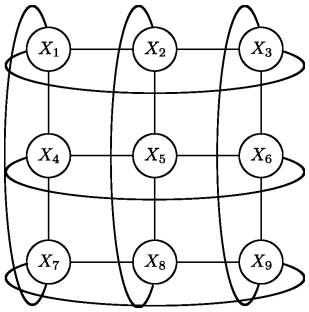

Fig. 6. 3×3 grid structure L showing the interactions between spins for a 2-D Ising spin glass with periodic boundaries. Each edge has an associated strength J_{ij}.

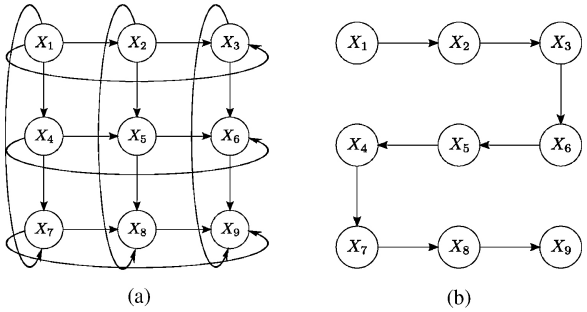

Fig. 7. Fixed structural models for 2-D Ising spin glass. (a) Dense structure. (b) Bivariate structure.

plete perspective about the EDA dynamics. For instance, the phenomenon we have described in Fig. 3, where the entropy initially tends to zero but later recovers, indicates that stopping criteria based on the entropy (e.g., [34]) should take this type of behavior into account to avoid early termination of the EDA.

V. EDA Behavior Solving Gaussian Ising

A. 2D Ising Spin Glass Description

Ising spin glass is an optimization problem which has been solved and analyzed in different works related to EDAs [15], [36], [48]. A classic 2-D Ising spin glass can be simply formulated. The set of variables $\boldsymbol{X}$ is seen as a set of n spins disposed on a regular 2-D grid L with $n = l \times l$ sites and periodic boundaries (see Fig. 6). Each node of L corresponds to a spin X_i and each edge (i, j) corresponds to a coupling between X_i and X_j. Thus, each spin variable interacts with its four nearest neighbors in the toroidal structure L. Moreover, each edge of L has an associated coupling strength J_{ij} between the related spins.

The target is, given couplings J_{ij}, to find the spin configuration that minimizes the energy of the system computed as

$$E(\boldsymbol{x}) = -\sum_{(i,j)\in L} x_i J_{ij} x_j - \sum_{i\in L} h_i x_i \tag{9}$$

where the sum runs over all coupled spins. In our experiments we take $h_i = 0\ \forall i \in L$. The states with minimum energy are called *ground states*.

Depending on the range chosen for the couplings J_{ij} we have different versions of the problem. For the *Gaussian Ising* problem, the couplings J_{ij} are real numbers generated following a Gaussian distribution. A specified set J_{ij} of coupling defines a spin glass instance. We generated 100 Gaussian Ising instances using the spin glass ground state server.[2] The minimum energy of the system is also provided by this server.

B. Structures Related to the Problem

In order to create a dense structure for this problem, we reproduce the undirected graph L in the model, which represents all the interactions among the variables in the function, and direct the edges without creating cycles to obtain a Bayesian network. Starting from the first spin (variable X_1) we give a westward and southward direction to the edges as can be seen in Fig. 7(a).

We are aware that for this problem, the directions given to the arcs could modify the behavior of EBNA, due to the different independence relations introduced in the Bayesian network [30]. However, as previously mentioned, the information introduced in the structural model is directly obtained from the formulation of the problem. In this sense, the direction of the arcs do not follow a specific criterion.

The second structure is a simple model which connects all variables using a chain. This structure introduces very few interactions related to the problem, as can be seen in Fig. 7(b).

C. Using Structural Learning

The evolution of the probability values in different situations is presented in Fig. 8. First of all, we note that when the population size given by bisection (m) is used, EBNA reaches the optimum in 470 runs out of 500, while with population size $m/2$ it decreases to 272. The proportion of successful runs with $m/2$ indicates that, in order to reach the optimum, the population size is less decisive in Gaussian Ising than in Trap5 (4 out of 50 successful runs with $m/2$).

Fig. 8(a) and (c) shows the probability values for successful runs with population sizes m and $m/2$, respectively. We can observe that the probabilistic behavior follows the same patterns as that in Trap5: 1) the probability of the optimum increases during the search, being the most probable solution at the end of the run, and 2) for the population size given by bisection, the curves for the MPS and the optimum are closer than for $m/2$. Nonetheless, it can be seen that the population size had a larger impact in Trap5 [Fig. 2(a) and (c)]. In that problem, the difference between the probability of the optimum and the highest in the distribution had a more emphasized change when the population size was varied (it was reflected in the number of successful runs). Moreover, while in Trap5 the probability of the optimum was very close to the highest from the first generations with population size m, in Gaussian Ising both probability curves keep a visible distance throughout the generations. Last, for Gaussian Ising, the runs do not reach the optimum [bars on the top of the charts in Fig. 8(a) and (c)] until their probability value exceeds approximately the threshold of -20 in logarithmic scale. We note that for Trap5, this threshold was much higher (-10). These particular differences in the probability values between

[2]Available at http://www.informatik.uni-koeln.de/ls_juenger/index.html.

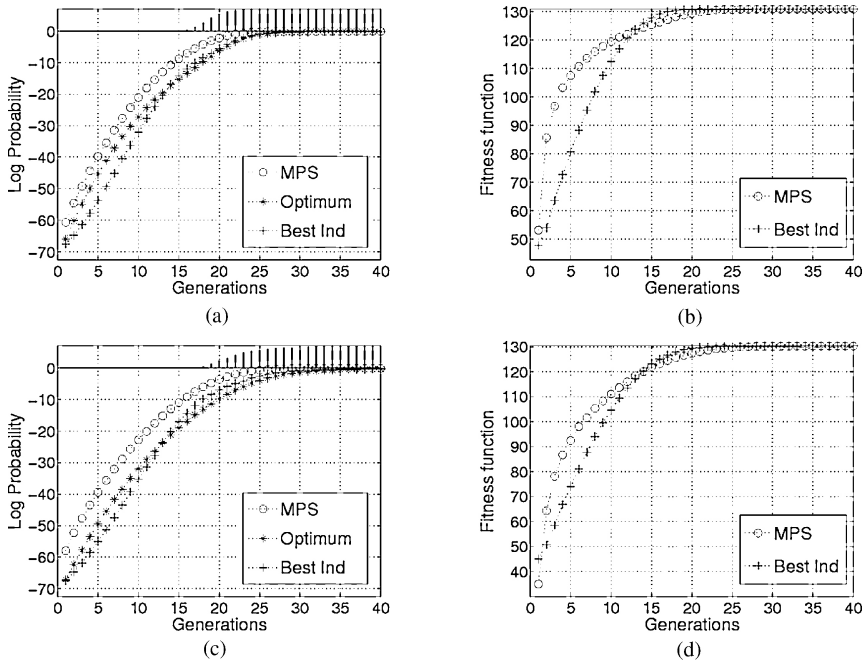

Fig. 8. Successful runs when EBNA is applied to Gaussian Ising using Algorithm B. We have 470 out of 500 successful runs with population size m and 272 out of 500 with population size $m/2$. (a) Population size m. (b) Population size m. (c) Population size $m/2$. (d) Population size $m/2$.

both problems could be due to the characteristics of the respective landscapes.

The analysis of the function values shown in Fig. 8(b) and (d) supports the previous discussion. As in Trap5, the difference in function value between the MPS and the best individual was bigger with population size m than with $m/2$. However, in this case, since the population size is less influential, the difference between curves is less marked than in the previous problem.

D. Using Fixed Structures

As in the previous problem, in this section we present the results for the population size m given by bisection. First of all, we note that the dense structure does not always reach the optimum as in Trap5. In particular, we achieved 283 out of 500 successful runs. Although the behavior of this structure does not outperform the behavior of EBNA with Algorithm B, its introduction in the algorithm has considerable consequences.

In Fig. 9(a), we report results of the analysis of the probability values and function values. In this case, the probability of the optimum is not the highest in the distribution during the search as in Trap5 [Fig. 4(a)]. However, the distance between both probability curves is smaller than with structural learning except for the last generations. This behavior is probably influenced by the criterion for directing the arcs. Depending on the instance, one selected direction could have important effects in the probability of the optimum. An example of this can be seen in Fig. 11.

In Fig. 9(c), we report the probability values when EBNA does not reach the optimum. We observe that the probability of the optimum is close to the highest in the distribution during the first generations and reaches values up to -20 before decreasing. In general, when the dense structure is introduced in the algorithm, the probability of the optimum in unsuccessful runs has higher values during the search than in previous scenarios. This knowledge about the probability of the optimum is important to understand EDAs and improve them.

Regarding the function values, as can be seen in Fig. 9(c), the MPS is close to the optimum from the beginning as in the case of Trap5. Moreover, this behavior remains constant in all of the instances analyzed. Again, this fact shows that using structures related with the interactions of the problem presents promising properties that deserve a specific study.

When we reduce the amount of information in the structural model, the effects in the algorithm are not only seen in the number of successful runs but also in the analysis. In this problem, when the bivariate structure of the chain is introduced, we only obtain 29 out of 500 successful runs. Looking at Fig. 10 we can see that, both in successful and unsuccessful runs, respectively, the probability of the optimum is more distant than in the corresponding previous scenarios from the highest probability. The low accuracy of the information about the problem that the probabilistic models contain is also reflected in the function values shown in Fig. 10(c). Although the MPS has a slightly higher function value than the best individual at the beginning of the run, the MPS is lower than best individual in the rest of the run.

To conclude this part of this paper, we would like to point out an interesting relationship between the probability and function values. In unsuccessful runs, the probability of the optimum always starts to decrease a few generations after the function values of the best individual reaches the values of the MPS. As the behavior of the function values is similar in successful and unsuccessful runs, this could suggest that the cross between both curves indicates a critical moment in the search.

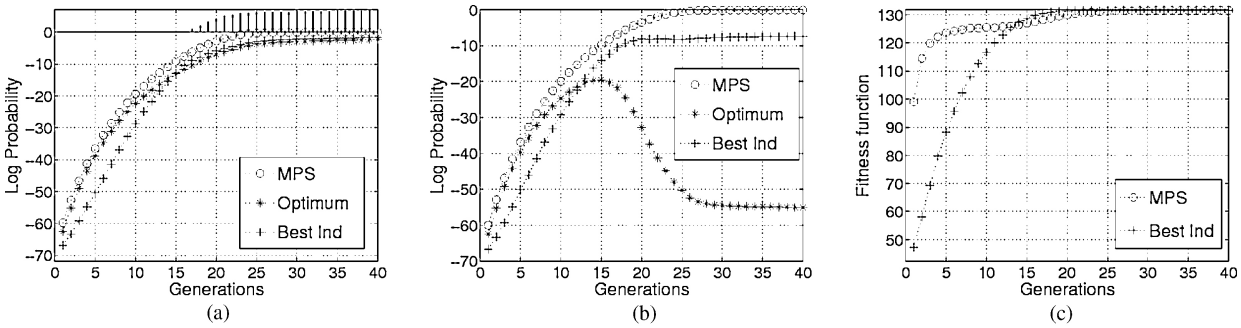

Fig. 9. Probability values and function values when EBNA is applied to Gaussian Ising using the dense structure with population size m. We have 283 out of 500 successful executions. (a) Successful runs. (b) Unsuccessful runs. (c) Successful runs.

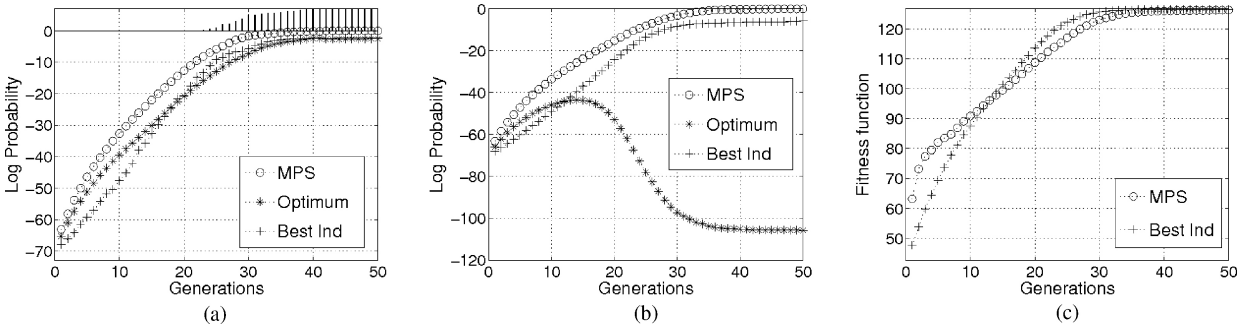

Fig. 10. Probability values and function values when EBNA is applied to Gaussian Ising using the bivariate structure with population size m. We have 29 out of 500 successful executions. (a) Successful runs. (b) Unsuccessful runs. (c) Successful runs.

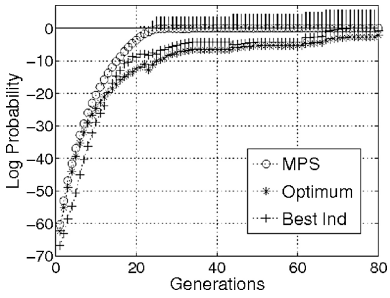

Fig. 11. Logarithm of the probabilities in successful runs when EBNA is applied to a particular instance of Gaussian Ising using the dense structure. For this instance, we have 10 out of 50 successful runs.

VI. EDA Behavior Solving ±J Ising

A. ± J Ising Description

As explained in Section V, the main difference between both versions of 2-D Ising spin glass is the range of values chosen for the couplings J_{ij}. In the present problem, the couplings J_{ij} are randomly set to either +1 or −1 with equal probability. This version, that will be called $\pm J$ *Ising*, could have different spin configurations that reach the ground state (lowest energy) and therefore many optimal solutions may arise. As in the previous case, 100 $\pm J$ Ising instances were generated using the spin glass ground state server.[3] This server also provided the value of the minimum energy of the system. As far as the fixed structural models are concerned, we use the same structures as in Gaussian Ising.

[3]Available at http://www.informatik.uni-koeln.de/ls_juenger/index.html.

B. Using Structural Learning

The analysis of problems with several optima reveals important changes in the internal behavior of the algorithm. Although the number of successful runs both with population size m (466 out of 500) and $m/2$ (226 out of 500) is very similar to Gaussian Ising, clear differences appear in the probability values. Fig. 12(a) shows that the probabilities assigned to the reached optima increase together during the generations. We have observed that the probability of the MPS does not tend to 1 (0 in logarithmic scale) as in unimodal problems. These facts indicate that the probability distribution is shared out among different optimal solutions in the last generations. This is verified by the accumulated entropy of the population (Fig. 14) which is greater than 0 at the end of the run. In Fig. 13, we illustrate the specific probability values of the MPS. We have seen that the MPS converges to higher probability values with $m/2$ because in this case, the average number of optimal solutions at the end of the run is lower than with m. In the same figure, we can see that this situation is repeated for unsuccessful runs [analysis shown in Fig. 12(b)]. This indicates that, although the optimum is not found, the algorithm reaches different solutions at the end of the run. The results for the accumulated entropy (see Fig. 14) confirm this behavior. An interesting behavior in this problem is that the MPS reaches higher probability values in successful runs than in unsuccessful ones (see Fig. 13). This is another issue that would deserve a specific analysis.

In Fig. 12(c), we show the function values for the MPS and the best individual. We can see how the best individual clearly exceeds the MPS after generation 12. This marked

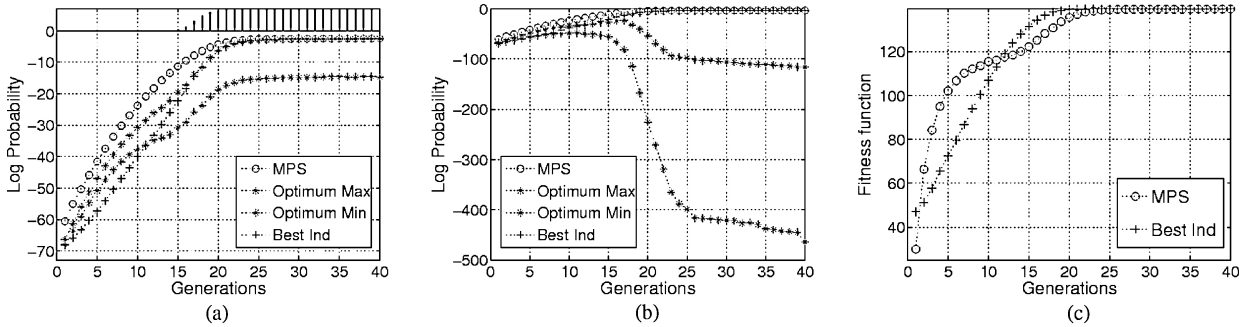

Fig. 12. Probability values and function values when EBNA is applied to $\pm J$ Ising using Algorithm B with population size m. We have 466 out of 500 successful runs with population size m and 226 out of 500 with population size $m/2$. On average, EBNA has reached 126 different optimal solutions at the end of the run with m and 55 with $m/2$. (a) Successful runs. (b) Unsuccessful runs. (c) Successful runs.

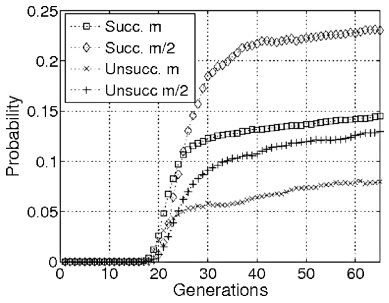

Fig. 13. Different curves of probability for the MPS when EBNA is applied to $\pm J$ Ising using Algorithm B. The curves correspond to successful runs with population size m (Succ. m), successful runs with $m/2$ (Succ. $m/2$), unsuccessful runs with m (Unsucc. m), and unsuccessful runs with $m/2$ (Unsucc. $m/2$).

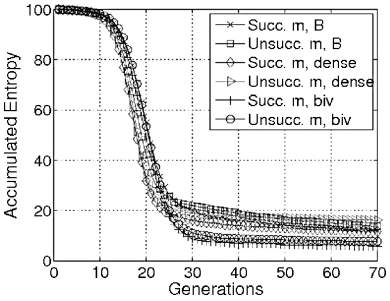

Fig. 14. Accumulated entropies of the population when EBNA solves $\pm J$ Ising.

fluctuation of the MPS occurs in the same generations when the probabilities of the optima slightly separate from the highest in the distribution in successful runs [Fig. 12(a)]. Moreover, the probabilities of the optima start to decrease in unsuccessful runs after the cross between the curves of function values. This supports the idea that this phase of the search is critical in order to reach the optimum.

C. Using Fixed Structures

In Fig. 15(a), we report the analysis of the probability values when the dense structure is introduced and the population size given by bisection is used. In the first generations, we can observe that the probabilities for the optima and the MPS are especially close. We also note that the maximum probability assigned to the set of optima is very close to the MPS during the search. According to the experiments, in this problem, the influence of the selected direction for the arcs is less dramatic than in Gaussian Ising. In fact, EBNA reaches the optimum in 463 out of 500 runs against 283 out of 500 in Gaussian Ising. This indicates that, depending on the problem, the same amount of information introduced in the structural model can have a different impact both on the probability values and the performance of the algorithm. It could depend on properties of the search space such as multimodality.

In Fig. 15(b) and (c), we show the results obtained when EBNA uses the bivariate structural model. In this case, we have 105 out of 500 successful executions. This is a clear improvement in the performance of the algorithm with regard to Gaussian Ising in this same scenario. This enhanced performance is reflected in the analysis of the function values [Fig. 15(c)]. In the first generation the MPS is clearly better than the best individual in the population. Moreover, at the beginning of the run, its difference is even more noticeable than in the case of EBNA using Algorithm B.

VII. EDA Behavior Solving Max-SAT

A. Max-SAT Description

The last problem in our analysis is the maximum satisfiability or Max-SAT problem, which has been often used in different works about EDAs [36], [37]. Put simply, given a set of Boolean variables $\boldsymbol{X}$ and a Boolean expression ϕ, SAT problem asks if there is an assignment $\boldsymbol{x}$ of the variables such that the expression ϕ is satisfied. In a Boolean expression, we can combine the variables using Boolean connectives such as $\wedge$ (logical and), $\vee$ (logical or), and $\neg$ (negation). An expression in the form x_i or $\neg x_i$ is called a literal.

Every Boolean expression can be rewritten into an equivalent expression in a convenient specialized style. In particular, we use the conjunctive normal form (CNF) $\phi = \bigwedge_{i=1}^{q} C_i$. Each of the q C_js is the disjunction of two or more literals which are called clauses of the expression ϕ. We work with clauses of length $k = 3$. When $k \geq 3$, the SAT problem becomes NP-Complete [43]. An example of a CNF expression with five Boolean variables X_1, X_2, X_3, X_4, X_5 and three clauses would be $\phi = (x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (x_1 \vee \neg x_4 \vee \neg x_2)$.

The Max-SAT problem has the same structure as SAT, but the result, for an assignment $\boldsymbol{x}$, is the number of satisfied

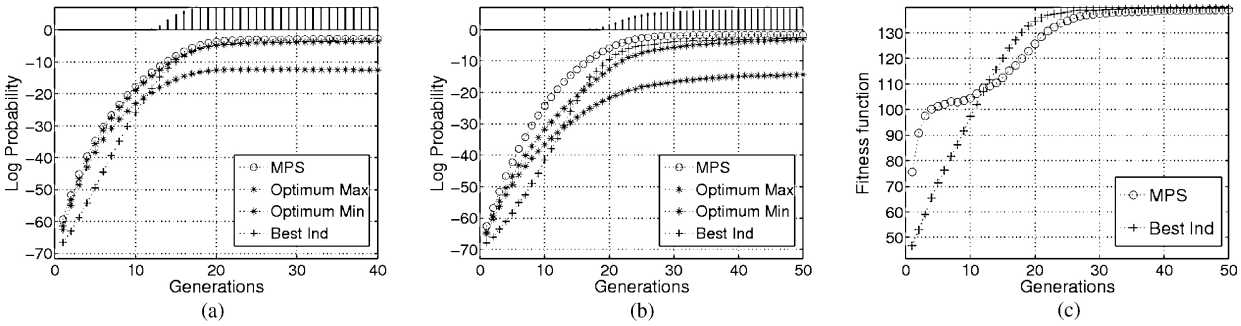

Fig. 15. Probability values and function values when EBNA is applied to $\pm J$ Ising using fixed structures with population size m. (a) Probability values for the dense structure. We have 463 out of 500 successful runs and, on average, EBNA has reached 138 different optimal solutions at the end of the runs. (b) Probability values for the bivariate structure. We have 105 out of 500 successful runs and, on average, EBNA has reached 47 different optimal solutions at the end of the runs. (c) Function values for the bivariate structure.

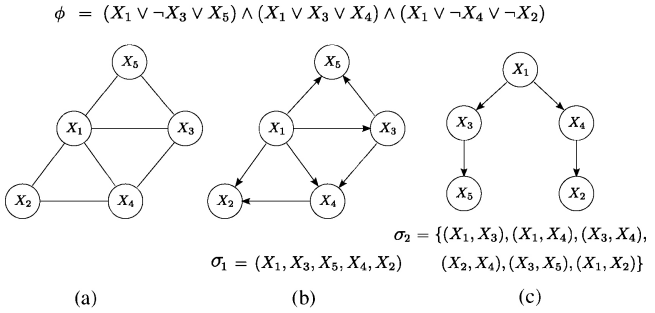

Fig. 16. Structures for MAX-SAT given a SAT expression ϕ. (a) Related undirected graph. (b) Related Bayesian network where σ_1 is the ancestral order. (c) Related tree structure where σ_2 is an order to add edges in the tree.

clauses instead of a truth value. In order to solve Max-SAT, the assignment for $\boldsymbol{X}$ that maximizes the number of satisfied clauses must be found. Thus, the optimization function can be written as

$$f_{Max-SAT}(\boldsymbol{x}) = \sum_{i=1}^{q} \phi(C_i) \tag{10}$$

where each clause C_i of three literals is evaluated as a Boolean expression that returns 1 if the expression is *true* or 0 if it is *false*. Since C_i is a disjunction, it is satisfied if at least one of its literals is *true*. The variables of $\boldsymbol{X}$ can overlap arbitrarily in the clauses.

Particularly, we work with the Uniform Random-3-SAT problems obtained from the SATLIB [49] repository. All the instances used are satisfiable. The presented results comprise 100 instances of 100 variables and 430 clauses. It is important to note that there could be several assignments for $\boldsymbol{X}$ that satisfy all clauses and therefore, this problem could have different optimal solutions.

B. Structures Related to the Problem

As different Max-SAT instances have different interactions among variables, a particular structure for each instance is needed. In order to create the dense structure, we join the variables belonging to the same clause C_i with edges. This step is illustrated in the example of Fig. 16(a) where a SAT formula is proposed. Now, in order to create a Bayesian network structure, we must direct the edges without creating cycles. In order to do this, we use an ancestral order which tries to minimize the number of parents per variable. Thus, the variables are ordered from the highest to the lowest number of overlaps in the clauses of the SAT instance. This type of structure is illustrated in Fig. 16(b) where σ_1 is the defined ancestral order. However, obtaining the MPS for such dense Bayesian network would be unfeasible due to the size of the cliques (up to 70 variables). Therefore, we were forced to reduce the complexity of the structure by deleting some edges. The high density of the interactions between variables in Max-SAT only allows us to work with two parents per variable. Thus, for each variable we select the two parents that correspond with the most frequent interactions with the child obtained from the clauses.

To create the bivariate structure, in this case a tree, we have followed a procedure similar to the Chow-Liu algorithm [50]. In Fig. 16(c), we illustrate a possible final result for a particular SAT formula. First, we create an order σ_2 for pairs of variables, related to the number of times that each couple of variables appear together in the SAT clauses, from the highest to the lowest. This is the scoring criterion for the arcs. Starting with an empty structure and following such an order, at each step we add an undirected edge without creating cycles. If there are ties, the selection is random. At the end of the procedure, the root of the tree is the most over-lapped variable in the SAT formula taken from the most frequent couple.

C. Using Structural Learning

In general, the analysis of EBNA when it is applied to Max-SAT shows similar behavior patterns to $\pm J$ Ising. However, as we previously discussed, each problem provides particular nuances to the analysis. For Max-SAT, we only show the results when EBNA uses the population size given by bisection because this parameter has a lower impact on the algorithm. In this problem, EBNA is only able to obtain bisection sizes for 20 out of 100 instances. In those instances where we do not have a bisection size, we use the maximum population size allowed in the experimentation (2^{14}). In order to analyze unsuccessful runs in instances for which EBNA is not able to reach the optimum, we introduced 350 optimal configurations obtained by a specific solver[4] for Max-SAT.

[4]Available at http://minisat.se.

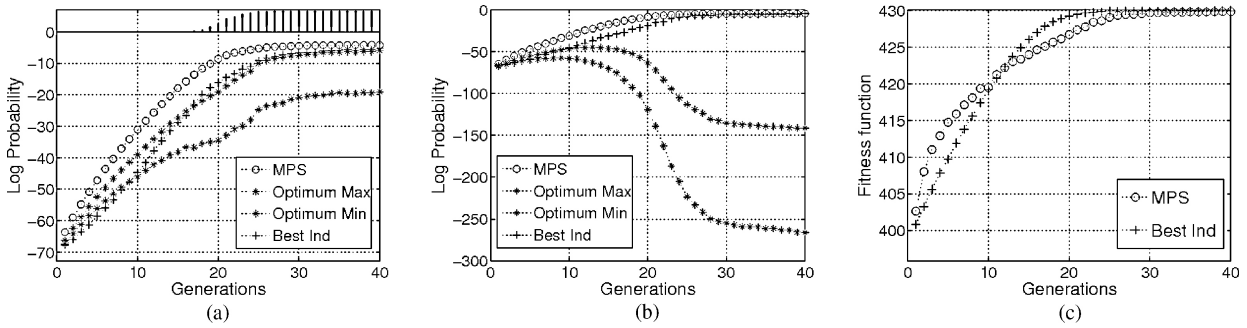

Fig. 17. Probability values and function values when EBNA is applied to Max-SAT using Algorithm B with population size m. We have 95 out of 500 successful executions. On average, EBNA has reached 1452 different optimal solutions at the end of the runs. (a) Successful runs. (b) Unsuccessful runs. (c) Successful runs.

As we can see in Fig. 17(a), the probability assigned to the set of optima is far from the highest in the distribution and this is reflected in the number of successful runs, in this case, 95 out of 500. Throughout this paper, we have seen that the more distant the MPS and the optima are, the lower the performance in terms of ratio of successful runs is. In this problem, the curves of probability indicate a lower exponential growth than in the rest of the problems. In fact, the different optimal solutions start to be reached in later generations [bars on the top of Fig. 17(a)]. We also note that the MPS reaches low probability values at the end of the runs (approximately, 0.05 for Max-SAT against 0.15 for $\pm J$ Ising in the same scenario). This is due to the high number of optima that EBNA is finding. Particularly, on average, we have 1452 different optimal solutions at the end of the runs.

In unsuccessful runs [Fig. 17(b)], the probabilities of the optima reach much lower maximum values than in the rest of the problems in the same scenario. Regarding the function values [Fig. 17(c)], although the MPS slightly outperforms the values of the best individual at the beginning of the run, this last solution is better than the MPS during a noticeable number of generations. Once again, the probabilities assigned to the optima start to decrease in unsuccessful runs some generations after the curve of the best individual crosses the curve of the MPS. The high number of optimal solutions, the great distance in probability between the MPS and the optima and the low quality of the function values of the MPS, reflect the hardness of this problem for EBNA.

D. Using Fixed Structures

In Fig. 18, we provide the probability values for successful runs. Although the dense structures only have a maximum of two parents per node, we can see the influence of this type of structure in the analysis. If we compare the dense structure with Algorithm B, we see that not only the maximum probability assigned to the set of optima is closer to the highest probability in the distribution, but also there is a lower difference in probability among the curves shown in the chart. Nevertheless, in all cases the optimum starts to be reached when the maximum probability of the optima has the value of âˆ’20 approximately. In contrast with structural learning, with fixed structures, EBNA reaches a lower number of optimal solutions at the end of the run, and this fact is also reflected in the final probability values. In this problem, the analysis shows a very similar behavior for both fixed structures.

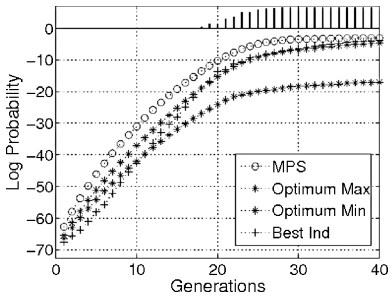

Fig. 18. Successful runs when EBNA is applied to Max-SAT using the dense structure with population size m. We have 31 out of 500 successful runs. On average, EBNA has reached 400 different optimal solutions at the end of the run.

VIII. Related Work

In [41], an analysis of the probability assigned by EDAs to the optimum solution is carried out for the Boltzmann EDA (BEDA) and factorized distribution algorithms that use valid and invalid factorizations. The analysis of the probabilities, which was carried out for a toy example, served to illustrate that, under the infinite population assumptions made by BEDA, the use of a valid factorization is a sufficient but not necessary condition for a steady increase, until convergence, of the probability given by BEDA to the optimum. Our work can be seen as an extension of the work presented in [41] in the sense that we investigate the probabilities of EDAs that apply in structural and parametric learning of a more complex class of models and across a range of different problems. We also provide a method to exactly determine the most probable solution given by the model.

Most of the research done concerning the models learned by EDAs based on Bayesian networks [19], [20], [51] has focused on structural descriptors of the networks, and specifically on the type (i.e., correct or spurious) and number of the network edges [11], [14]–[16], [22], [52]. The analysis of the Bayesian network edges learned by EDAs has allowed us to study the effect of the selection and replacement procedures [15], [16] as well as the learning method [14], [22], [52] in the accuracy of the models learned by EDAs and the efficiency of these algorithms. A more recent work [52] considers the likelihood given to the selected set during the model learning

step as another source of information about the behavior of the algorithm. In this case, not only the structure but also the probabilities are taken into consideration when computing the model descriptor. Nevertheless, none of the previously mentioned papers uses the probabilities given by the models to some distinguished solutions (e.g., most probable explanation, known optimum, and so on) as a means to reveal information about EDAs. In no case are there references to the most likely solution that could be sampled from the learned model.

For EDAs that use Markov models [53]–[55], different issues related to the relationship between the fitness function and the probabilistic models learned by EDAs have been investigated. Relevant to the work presented in this paper, is the use of the models learned by the distribution estimation using Markov network algorithm (DEUM) [53], [55], [56] as predictors of the fitness function.

In [13], the product moment correlation coefficient between the Markov model learned by DEUM and the fitness function is used to measure the quality of the model as a fitness function predictor. For a given solution, the prediction is the value given by the Markov model to the solution. The quality of the model is measured using the correlation computed from samples of the search space. Furthermore, the prediction accuracy of Markov models with different structural complexity is investigated for different selection strategies and population sizes.

A substantial difference between the work presented in [13] and the results introduced in this paper is that the analysis of the prediction given by the models is constrained to the solutions taken from the selected population or random samples. The most probable explanation given by the model is not computed. Another difference is that the evolution of the models throughout the generations is not analyzed. By computing the most probable individual given by the model at each generation, we are able to obtain a dynamic view of the quality of the probability model.

IX. Conclusion

In this paper, we analyzed EDAs from a quantitative point of view in order to better understand their internal behavior. Through the recording of probability and function values for a set of distinguished solutions during the search, we directly studied the probability distributions generated by this type of algorithms. More specifically, the proposed analysis has allowed us to investigate basic open issues raised in Section I, whose study entails a deeper understanding and development of EDAs. Now, we return to these questions, providing the new knowledge that we obtained throughout this paper.

1) How does the probability assigned to the optimal solution by the probability distributions evolve during the search?

We can distinguish different scenarios depending on the number of optimal solutions of the function to be optimized and the success of the search. In general, in order to reach an optimal solution, its probability must exceed a certain threshold which can vary depending on the intrinsic characteristics of the problem. On the one hand, when EBNA is applied to unimodal problems (Trap5 and Gaussian Ising) and the optimal solution is found, its probability continuously increases until it reaches the value of 1. One exception is function Trap5 and the bivariate structure where the probability of the optimum decreases at the beginning of the run and it increases in the last generations.

On the other hand, when EBNA successfully solves multimodal problems ($\pm J$ Ising and Max-SAT), it is able to reach a subset of the optimal solutions and their probability values also increase during the search. In these problems, the probability is distributed among different solutions at the end of the run (note that the number of generations is limited). Thus, the non-convergence to 1 of the probability values of the MPS or the best individual of the population (both probability curves always rise simultaneously) reflects the multimodality of the function. Moreover, these probability values are lower when the algorithm reaches a higher number of optima. This finding can be used to detect multimodality when an unknown problem needs to be faced.

In unsuccessful runs, the probability of the optimum always has a similar pattern. At the beginning of the run, it increases together with the probability of the MPS and the best individual of the population. However, after a certain generation, before reaching a specific probability threshold, it decreases rapidly.

Both the probability of the MPS and the best individual of the population in logarithmic scale, accurately show how that the algorithm converges as the generations advance. Therefore, by monitoring the probability of the best individual, it would be possible to know the speed of convergence of the algorithm and bring forward a premature convergence. According to this, modifications in the replacement technique could be performed in order to regulate the diversity of the population. This information could also be useful in order to distinguish between exploration and exploitation phases. Thus, we could stop the search at the right time (before the probability of the optimum starts to decrease) and take advantage of the information contained in the probabilistic model by using exploitation techniques.

The population size also influences the probability assigned to the optimum during the search and this is reflected in the number of successful runs. When the population size given by bisection is used, the probability values for the optimum tend to be closer to the highest in the distribution and, in most of the cases, this is beneficial in order to solve the problem.

2) How does the accuracy of the information about the problem contained in the structural model influence the internal behavior of EDAs?

The results support the conclusion obtained in [15] regarding the difficulty of creating adequate probabilistic models by hand even with complete knowledge of the problem structure. However, the quantitative analysis of the models reveals that the use of information about the problem has an important impact in the internal behavior of the algorithm.

In particular, we provide two clear conclusions. First, when we are able to introduce all the interactions between the variables of the problem, the probability of the optimum tends

to be closer to the highest probability in the distribution. Secondly, when we introduce this information, the function value for the MPS is very close to the optimum from the beginning of the run. However, despite these favorable properties, these types of models do not always perform satisfactorily. The experimental results indicate that the PLS sampling method (one of the most widely used) does not extract all the valuable information contained in the probabilistic models. For this reason, in order to take advantage of both the high probability assigned to the optimum and the high quality function values of the MPS, the use of a sampling based on exact or even approximate belief propagation techniques [57], [58] could be beneficial. Another reason we point out for such non-constant behavior is the direction assigned to the arc, which conditions the order the variables will be sampled by the PLS technique. A possible solution for this issue could be to, according to a given score, look for the best direction for the arcs at each step of the algorithm.

When the information about the problem that the probabilistic model contains is reduced, the probability of the optimum is more distant from the highest in the distribution. Moreover, the function value of the best individual is closer to the MPS in the first generations. These facts justify the poor performance of the algorithm in these cases.

3) How does the function value for the most probable solution evolve during the search?

The function value for the MPS always increases during the search until it stabilizes in the last generations. At the beginning of the run, it is usually better than the best individual in the population.

As we previously said, the difference between the function values of the MPS and the best individual increases when a dense structure is used. Another interesting observation is that this difference also reflects the impact that the population size has on a particular problem. Thus, when we increase the population size in order to solve Trap5, both the difference between the function values analyzed and the number of successful runs clearly increases. However, for Max-SAT, increasing the population size hardly influences the curves of function values and the performance of the algorithm. Therefore, by analyzing the MPS and the best individual with different population sizes, we can predict, without additional information about the problem, if increasing this parameter will be useful in order to obtain better results or if we need to look for other solutions to improve the performance of the algorithm.

By using these function values, we believe that it is also possible to identify different phases of the search. According to the results, in unsuccessful runs, the probability of the optimum starts to decrease shortly after the function value of the best individual outperforms the function value of the MPS. Moreover, the optimum is never reached before this event. It could be used to identify the end of the exploration stage and avoid a premature convergence.

In summary, the difference in function value between the MPS and the best individual could be used: 1) to improve the setup of EDA parameters; 2) to measure the quality of the information introduced about the problem in the model; 3) to measure the quality of sampling methods; and 4) to detect critical phases in the search. Finally, the analysis carried out in this paper has become useful in order to learn about different aspects of the algorithm and propose improvement solutions. We believe that similar approaches to analyze EDAs can be especially useful for other EDA practitioners both in fundamental research and in real problem applications.

X. Future Work

There are a number of trends where it is worth extending the results presented in this paper.

A direct extension of this paper is to reproduce the proposed analysis in problems with different characteristics as, for example, non-binary discrete problems. Another line of future research is to study the influence, in the descriptors of the probabilistic models introduced in this paper, of different selection operators or replacement strategies. Particularly, the influence of techniques to avoid loss of diversity such as niching [59] or restricted tournament selection [60] deserves a deep study. Similarly, the influence on our descriptors of the model learning algorithm used (e.g., exact versus approximate learning of Bayesian networks [14]) as well as the scoring metric to estimate the quality of the networks (e.g., Bayesian-Dirichlet metric [61] or Bayesian information criterion [26]) is worth further investigation. The introduction of alternative descriptors of the EDA behavior such as the entropy of the Bayesian networks would be interesting.

Some of the ideas collected in the conclusions can be used in the development of adaptive EDAs [62]. For example: 1) by using the probability value of the best individual of the population to measure the convergence of the algorithm and make online decisions about the replacement technique, and 2) by using the relation between the MPS and the best individual in order to self-adjust the population size or to use an adequate sampling method during the search. Furthermore, we could study the use of approximate techniques such as loopy belief propagation [18] to calculate the MPS, because in general, its exact computation is constrained by the size of the problem.

Finally, taking into account the constant patterns observed in the behavior of the probabilities, we think it could be possible to theoretically model the relation between probability curves and parameters such as the number of variables and the population size. In summary, we believe that these experimental results will help to further develop this type of theoretical model.

Acknowledgment

The authors would like to thank the anonymous reviewers whose comments have contributed to improve this paper.

References

[1] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I: Binary parameters," in *Proc. 4th PPSN*, LNCS 1141. 1996, pp. 178–187.

[2] P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Boston, MA: Kluwer, 2002.

[3] M. Pelikan, *Hierarchical Bayesian Optimization Algorithm. Toward a New Generation of Evolutionary Algorithms* (Series Studies in Fuzziness and Soft Computing). Berlin, Germany: Springer, 2005.

[4] P. A. Simionescu, D. Beale, and G. V. Dozier, "Teeth-number synthesis of a multispeed planetary transmission using an estimation of distribution algorithm," *J. Mech. Des.*, vol. 128, no. 1, pp. 1–12, 2007.

[5] R. Armañanzas, I. Inza, R. Santana, Y. Saeys, J. L. Flores, J. A. Lozano, Y. V. de Peer, R. Blanco, V. Robles, C. Bielza, and P. Larrañaga, "A review of estimation of distribution algorithms in bioinformatics," *BioData Min.*, vol. 1, no. 6, pp. 1–12, Sep. 2008.

[6] R. Santana, P. Larrañaga, and J. A. Lozano, "Protein folding in simplified models with estimation of distribution algorithms," *IEEE Trans. Evol. Comput.*, vol. 12, no. 4, pp. 418–438, Aug. 2008.

[7] B. Yuan, M. E. Orlowska, and S. W. Sadiq, "Finding the optimal path in 3-D spaces using EDAs: The wireless sensor networks scenario," in *Proc. Adap. Nat. Comput. Algorithms 8th Int. Conf. ICANNGA*, Apr. 2007, pp. 536–545.

[8] R. Santana, P. Larrañaga, and J. A. Lozano, "Research topics on discrete estimation of distribution algorithms," *Memetic Comput.*, vol. 1, no. 1, pp. 35–54, 2009.

[9] Q. Zhang and H. Mühlenbein, "On the convergence of a class of estimation of distribution algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 127–136, Apr. 2004.

[10] M. Pelikan, K. Sastry, and D. E. Goldberg, "Sporadic model building for efficiency enhancement of the hierarchical BOA," *Genetic Program. Evol. Mach.*, vol. 9, no. 1, pp. 53–84, 2008.

[11] M. Hauschild and M. Pelikan, "Enhancing efficiency of hierarchical BOA via distance-based model restrictions," Missouri Estimation Distribution Algorithms Lab., Univ. Missouri, St. Louis, MEDAL Rep. 2008007, Apr. 2008.

[12] E. Bengoetxea, "Inexact graph matching using estimation of distribution algorithms," Ph.D. dissertation, Dépt. Traitement du Signal et des Images, Ecole Nationale Supérieure des Télécommunications, Paris, France, 2003.

[13] S. Brownlee, J. McCall, Q. Zhang, and D. Brown, "Approaches to selection and their effect on fitness modeling in an estimation of distribution algorithm," in *Proc. CEC*, 2008, pp. 2621–2628.

[14] C. Echegoyen, R. Santana, J. Lozano, and P. Larrañaga, "The impact of exact probabilistic learning algorithms in EDAs based on Bayesian networks," in *Linkage in Evolutionary Computation*. Berlin/Heidelberg, Germany: Springer, 2008, pp. 109–139.

[15] M. Hauschild, M. Pelikan, K. Sastry, and C. Lima, "Analyzing probabilistic models in hierarchical BOA," *IEEE Trans. Evol. Computat.*, vol. 13, no. 6, pp. 1199–1217, Dec. 2009.

[16] C. F. Lima, M. Pelikan, D. E. Goldberg, F. G. Lobo, K. Sastry, and M. Hauschild, "Influence of selection and replacement strategies on linkage learning in BOA," in *Proc. CEC*, 2007, pp. 1083–1090.

[17] H. Mühlenbein and R. Höns, "The factorized distributions and the minimum relative entropy principle," in *Scalable Optimization Via Probabilistic Modeling: From Algorithms to Applications* (Series Studies in Computational Intelligence), M. Pelikan, K. Sastry, and E. Cantú-Paz, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 11–38.

[18] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[19] R. Etxeberria and P. Larrañaga, "Global optimization using Bayesian networks," in *Proc. 2nd Symp. Artif. Intell. (CIMAF)*, 1999, pp. 151–173.

[20] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "BOA: The Bayesian optimization algorithm," in *Proc. GECCO*, vol. 1. 1999, pp. 525–532.

[21] H. Mühlenbein and T. Mahnig, "FDA: A scalable evolutionary algorithm for the optimization of additively decomposed functions," *Evol. Computat.*, vol. 7, no. 4, pp. 353–376, 1999.

[22] C. Echegoyen, J. A. Lozano, R. Santana, and P. Larrañaga, "Exact Bayesian network learning in estimation of distribution algorithms," in *Proc. CEC*, 2007, pp. 1051–1058.

[23] C. Echegoyen, A. Mendiburu, R. Santana, and J. Lozano, "Analyzing the probability of the optimum in EDAs based on Bayesian networks," in *Proc. CEC*, 2009, pp. 1652–1659.

[24] M. Hauschild, M. Pelikan, K. Sastry, and D. E. Goldberg, "Using previous models to bias structural learning in the hierarchical BOA," Missouri Estimation Distribution Algorithms Lab., Univ. Missouri, St. Louis, MEDAL Rep. 2008003, 2008.

[25] W. Buntine, "Theory refinement on Bayesian networks," in *Proc. 7th Conf. Uncertainty Artif. Intell.*, 1991, pp. 52–60.

[26] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 7, no. 2, pp. 461–464, 1978.

[27] M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling," in *Proc. 2nd Annu. Conf. Uncertainty Artif. Intell.*, 1988, pp. 149–164.

[28] J. Gámez, "Abductive inference in Bayesian networks: A review," in *Advances in Bayesian Networks*. Berlin, Germany: Springer, 2004, pp. 101–120.

[29] S. E. Shimony, "Finding MAPs for belief networks is NP-hard," *Artif. Intell.*, vol. 68, no. 2, pp. 399–410, 1994.

[30] E. Castillo, J. M. Gutierrez, and A. S. Hadi, *Expert Systems and Probabilistic Network Models*. Berlin, Germany: Springer-Verlag, 1997.

[31] R. Dechter, "Bucket elimination: A unifying framework for reasoning," *Artif. Intell.*, vol. 113, nos. 1–2, pp. 41–85, 1999.

[32] K. Murphy, "The Bayes net toolbox for MATLAB," in *Proc. Interface Comput. Sci. Statist.*, vol. 33. 2001, pp. 1024–1034.

[33] A. Ochoa and M. R. Soto, "Linking entropy to estimation of distribution algorithms," in *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 1–38.

[34] J. Ocenasek, "Entropy-based convergence measurement in discrete estimation of distribution algorithms," in *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 39–50.

[35] R. Santana, C. Bielza, P. Larrañaga, J. A. Lozano, C. Echegoyen, A. Mendiburu, R. Armañanzas, and S. Shakya, "MATEDA: Estimation of distribution algorithms in MATLAB," *J. Statist. Softw.*, vol. 35, no. 7, pp. 1–30, 2010.

[36] M. Pelikan and D. E. Goldberg, "Hierarchical BOA solves Ising spin glasses and MAXSAT," in *Proc. GECCO*, 2003, pp. 1271–1282.

[37] S. Brownlee, J. McCall, and D. Brown, "Solving the MAXSAT problem using a multivariate EDA based on Markov networks," in *Proc. GECCO*, 2007, pp. 2423–2428.

[38] K. Deb and D. E. Goldberg, "Sufficient conditions for deceptive and easy binary functions," *Ann. Math. Artif. Intell.*, vol. 10, no. 4, pp. 385–408, 1994.

[39] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[40] E. Ising, "The theory of ferromagnetism," *Z. Phys.*, vol. 31, no. 1, pp. 253–258, 1925.

[41] H. Mühlenbein, T. Mahnig, and A. Ochoa, "Schemata, distributions and graphical models in evolutionary optimization," *J. Heuristics*, vol. 5, no. 2, pp. 213–247, 1999.

[42] F. Barahona, "On the computational complexity of Ising spin glass model," *J. Phys. A: Math. General*, vol. 15, no. 10, pp. 3241–3253, 1982.

[43] S. A. Cook, "The complexity of theorem-proving procedures," in *Proc. 3rd Annu. ACM Symp. Theory Comput.*, 1971, pp. 151–158.

[44] N. Friedman and Z. Yakhini, "On the sample complexity of learning Bayesian networks," in *Proc. 12th Conf. UAI*, 1996, pp. 274–282.

[45] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[46] U. B. Kjaerulff, "Triangulation of graphs: Algorithms giving small total state space," Dept. Comput. Sci., Aalborg Univ., Myrdalstræde, Denmark, Tech. Rep. R-90-09, Mar. 1990.

[47] A. Wright, R. Poli, C. Stephens, W. B. Langdon, and S. Pulavarty, "An estimation of distribution algorithm based on maximum entropy," in *Proc. Genetic Evol. Computat. Conf.*, 2004, pp. 343–354.

[48] M. Pelikan and A. K. Hartmann, "Searching for ground states of Ising spin glasses with hierarchical BOA and cluster exact approximation," in *Scalable Optimization Via Probabilistic Modeling: From Algorithms to Applications* (Series Studies in Computational Intelligence), M. Pelikan, K. Sastry, and E. Cantú-Paz, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 333–349.

[49] H. Hoos and T. Stutzle, "SATLIB: An online resource for research on SAT," in *Proc. SAT*, 2000, pp. 283–292.

[50] C. K. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. Inform. Theory*, vol. 14, no. 3, pp. 462–467, May 1968.

[51] H. Mühlenbein and T. Mahnig, "Evolutionary synthesis of Bayesian networks for optimization," in *Advances in Evolutionary Synthesis of Intelligent Agents*, M. Patel, V. Honavar, and K. Balakrishnan, Eds. Cambridge, MA: MIT Press, 2001, pp. 429–455.

[52] C. F. Lima, F. G. Lobo, and M. Pelikan, "From mating pool distributions to model overfitting," in *Proc. GECCO*, 2008, pp. 431–438.

[53] S. Shakya and J. McCall, "Optimization by estimation of distribution with DEUM framework based on Markov random fields," *Int. J. Automat. Comput.*, vol. 4, no. 3, pp. 262–272, 2007.
[54] R. Santana, "A Markov network based factorized distribution algorithm for optimization," in *Proc. 14th ECML-PKDD*, vol. 2837. 2003, pp. 337–348.
[55] S. Shakya, "DEUM: A framework for an estimation of distribution algorithm based on Markov random fields," Ph.D. dissertation, School Comput., Robert Gordon Univ., Aberdeen, U.K., 2006.
[56] S. Shakya, J. McCall, and D. Brown, "Using a Markov network model in a univariate EDA: An empirical cost-benefit analysis," in *Proc. GECCO*, 2005, pp. 727–734.
[57] A. Mendiburu, R. Santana, and J. A. Lozano, "Introducing belief propagation in estimation of distribution algorithms: A parallel framework," Dept. Comput. Sci. Artif. Intell., Univ. Basque Country, San Sebastián-Donostia, Spain, Tech. Rep. EHU-KAT-IK-11/07, Oct. 2007.
[58] C. F. Lima, M. Pelikan, F. G. Lobo, and D. E. Goldberg, "Loopy substructural local search for the Bayesian optimization algorithm," in *Proc. SLS*, vol. 5752. 2009, pp. 61–75.
[59] S. W. Mahfoud, "Niching methods for genetic algorithms," Ph.D. dissertation, Genetic Algorithm Lab., Univ. Illinois Urbana-Champaign, Urbana, IlliGAL Rep. 95001, May 1995.
[60] G. R. Harick, "Finding multimodal solutions using restricted tournament selection," in *Proc. ICGA*, 1997, pp. 24–31.
[61] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *J. Mach. Learn.*, vol. 9, no. 4, pp. 309–347, Oct. 1992.
[62] R. Santana, P. Larrañaga, and J. A. Lozano, "Adaptive estimation of distribution algorithms," in *Adaptive and Multilevel Metaheuristics* (Series Studies in Computational Intelligence), C. Cotta, M. Sevaux, and K. Sörensen, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 177–197.

Carlos Echegoyen received the B.S. degree in computer science from the University of the Basque Country, San Sebastián-Donostia, Spain, in 2007. He is currently pursuing the Ph.D. degree from the Intelligent Systems Group, University of the Basque Country.

His current research interests include evolutionary computation and probabilistic graphical models.

Alexander Mendiburu received the B.S. and Ph.D. degrees in computer science from the University of the Basque Country, San Sebastián-Donostia, Spain, in 1995 and 2006, respectively.

He is currently an Associate Professor with the Department of Computer Architecture and Technology, Intelligent Systems Group, University of the Basque Country. His current research interests include evolutionary computation, probabilistic graphical models, and parallel and distributed computing.

Roberto Santana received the B.S. degree in computer science and the Ph.D. degree in mathematics from the University of Havana, Havana, Cuba, in 1996 and 2005, respectively, and the Ph.D. degree in computer science from the University of the Basque Country, San Sebastián-Donostia, Spain, in 2006.

He is currently a Post-Doctoral Researcher with the Cajal Blue Brain Project, Technical University of Madrid, Madrid, Spain. His current research interests include evolutionary computation, probabilistic graphical models, neuroscience, and bioinformatics.

Jose A. Lozano (M'04) received the B.S. degree in mathematics and the B.S. degree in computer science from the University of the Basque Country, San Sebastián-Donostia, Spain, in 1991 and 1992, respectively, and the Ph.D. degree in computer science from the University of the Basque Country in 1998.

Since 2008, he has been a Full Professor with the University of the Basque Country, where he leads the Intelligent System Group. He is the co-author of more than 50 ISI journal publications and the co-editor of the first book published about estimation of distribution algorithms. His current research interests include machine learning, pattern analysis, evolutionary computation, data mining, metaheuristic algorithms, and real-world applications.

Prof. Lozano is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and a member of the Editorial Board of *Evolutionary Computation*, *Soft Computing*, and other three journals.