Mining Probabilistic Models Learned by EDAs in the Optimization of Multi-objective Problems

Roberto Santana
Universidad Politécnica de Madrid
Madrid, Spain
rsantana@ehu.es

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

Jose A. Lozano
CCIA, University of the Basque Country
San Sebastian, Spain
ja.lozano@ehu.es

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

ABSTRACT

One of the uses of the probabilistic models learned by estimation of distribution algorithms is to reveal previous unknown information about the problem structure. In this paper we investigate the mapping between the problem structure and the dependencies captured in the probabilistic models learned by EDAs for a set of multi-objective satisfiability problems. We present and discuss the application of different data mining and visualization techniques for processing and visualizing relevant information from the structure of the learned probabilistic models. We show that also in the case of multi-objective optimization problems, some features of the original problem structure can be translated to the probabilistic models and unveiled by using algorithms that mine the model structures.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Estimation of distribution algorithms, problem structure, probabilistic modeling, visualization

1. INTRODUCTION

Estimation of distribution algorithms (EDAs) [14, 19, 21] are evolutionary algorithms that use population of points and apply selection based on the fitness of the individuals. To this respect they are similar to genetic algorithms (GAs) [7, 10]. However, they replace traditional crossover operators used by GAs by the estimation and sampling of probabilistic models.

An important characteristic of EDAs is their capacity to learn a model of the problem during the search. Although the main use of the probabilistic model learned by EDAs is to sample new solutions, the model can be also used to capture previously unknown information about the problem structure. As a result, EDAs can be seen not only as an optimization algorithm but also as a simulation tool to produce information about the problem domain.

One of the factors that influences the extent to which the information contained in the probabilistic models is exploited is the class of techniques available to preprocess and visualize this information. The use of more sophisticated preprocessing and visualization techniques can contribute to increase the amount of information extracted from the models, to enhance the quality of this information, and help the user to transform this information into knowledge about the problem.

In this paper we present and discuss the application of methods for extracting, processing and visualizing relevant information from the structure of the probabilistic models learned. We show the way in which data preprocessing and visualization techniques can help to reveal and elaborate this information.

The information mined for the models can be applied in a variety of ways: e.g. to evaluate the model accuracy in capturing the original problem interactions, to improve the EDA and design local optimization algorithms, to classify the problem instances according to the structural patterns extracted from the learned models, etc. However, the focus of this paper is on expanding the set of current techniques employed to analyze the probabilistic models in EDAs. Currently, the investigation is mainly based on the independent analysis of the edge frequencies computed from the adjacency matrices that represent the structures of the probabilistic models.

We extend the analysis of interactions between variables to the analysis of interactions between edges, and present techniques that allow to visualize these interactions globally

GECCO'09, July 8–12, 2009, Montréal, Québec, Canada.

(hierarchical association of edges in the models) and dynamically (interactions between edges at different generations). To illustrate the usefulness of these techniques, we search for relationships between the original multi-objective satisfiability problem structure and the structural characteristics of the models learned by EDAs.

The paper is organized as follows: In the next section, EDAs are presented and the Bayesian network-based EDA used for the optimization of the multi-objective problems is explained. Section 3 introduces a possible general strategy to extract information from the models learned by EDAs. Previous work on the analysis of the probabilistic models learned in EDAs and discrete multi-objective problems treated with these algorithms is briefly reviewed in Section 4. Section 5 presents the multi-objective satisfiability problem and results obtained by analyzing the graphical models generated by EDAs in the solution of this problem. The conclusions of our paper and some lines for future research are given in Section 6.

2. AN EDA FOR MULTI-OBJECTIVE PROBLEMS

Distinctive features of EDAs are the type of probabilistic model, and the particular class of learning and sampling methods. The models may differ in the order and number of the probabilistic dependencies that they represent. A variety of learning and sampling techniques can be used in accordance to the type of representation and other characteristics of the optimization problem. In particular, there may be important differences between EDA implementations for single and multi-objective problems. Enforcing the population diversity needed to guarantee a good covering of the Pareto set is particularly important for multi-objective problems and specialized learning and sampling methods may be conceived to fulfil this goal.

The estimation of Bayesian networks algorithm (EBNA) [5] learns a Bayesian network from the database containing the selected individuals at each generation. We have adapted the EBNA algorithm to deal with multi-objective problems by modifying the selection step.

The selection method employed uses Pareto ranking selection where individuals are ordered according to the front they belong to. Individuals in the first front (non-dominated solutions) come first. Then individuals that are only dominated by those in the first front and so on. Within each front, they are ordered according to the average rank of their fitness functions. After all the population has been ordered, truncation selection of the T percentage of the population is done.

A pseudocode of EBNA is shown in Algorithm 1. The algorithm was implemented in Matlab using the MATEDA-2.0 software [23]. The learning and sampling steps of the Bayesian networks are implemented using the Matlab Bayes Net (BNT) toolbox [20]. The scoring metric used was the Bayesian metric with uniform priors, and each node was allowed a maximum number of 5 parents.

Algorithm 1: **EBNA**

1 $BN_0 \leftarrow (S_0, \boldsymbol{\theta}^0)$ where S_0 is an arc-less DAG, and $\boldsymbol{\theta}^0$ is uniform
2 $p_0(\mathbf{x}) = \prod_{i=1}^{n} p(x_i) = \prod_{i=1}^{n} \frac{1}{r_i}$
3 $D_0 \leftarrow$ Sample M individuals from $p_0(\mathbf{x})$ and evaluate them
4 t $\leftarrow 1$
5 **do** {
6 $D_{t-1}^{Se} \leftarrow$ Select N individuals from D_{t-1} using Pareto-ranking selection.
7 $S_t^* \leftarrow$ Use local search to find one network structure that optimizes scoring metric
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 M individuals from BN_t and evaluate them
11 } **until** Stop criterion is met

3. STAGES IN THE ANALYSIS OF MODELS

We propose the following steps for the analysis of the information contained in the models:

- Extraction of the structures.
- Mining of the structures.
- Visualization of the relevant features.

3.1 Extraction of the structures

In some probabilistic models such as Bayesian and Gaussian networks the identification of the model structure is straightforward. However, some transformation may also be needed. For instance, the direction of the arcs is usually disregarded when the structure of the Bayesian network is used for the analysis. Other models may require a step where more sophisticated techniques are employed to identify the structure. One example are correlation matrices used to represent interactions in continuous problems. In such models, some underlying structure is assumed to exist but it has to be recovered by means of some action, e.g. thresholding the correlation values according to some parameter.

3.2 Mining of the structures

Once the structure has been extracted, the next step is to mine it looking for some predefined characteristics or trying to identify salient features. The type of search will depend on whether we are trying to find clues about the behavior of the algorithm such as the capacity of the probabilistic model to represent the problem structure, the accuracy of the model learning algorithm to detect the interactions, etc. or whether we intend to reveal characteristic features about the problem domain such as the set of variables that interact, the hierarchical structure of the interactions, etc. Notice that although the structure mining step is related to the structure extraction step, in the former we are making a discrimination between the structures according to some properties. On the other hand, the extraction step can be also seen as the starting point for the more complex structure mining step.

There are several features that can be extracted from the model structures, among them, the following:

- Number of edges.
- Distribution of the vertex degree.

- Size of the maximum clique.
- Number of maximal cliques.
- Number and average size of connected components.
- Sets of interacting components.

These features can also support information about the complexity of the problems. For example, the size of the maximum clique has an important role in the complexity of the sampling procedures and the critical population size of EDAs. Other topological measures used to characterize graphs and networks [1] (e.g. centrality of the graph, clustering coefficient and average path length between vertices) could be used to unveil relevant information about the problem from the analysis of the structures.

There are different algorithms that allow to extract from the model structures the previously mentioned topological features of the graphs. Some of the algorithms may be very costly, as is the case with the methods used to find all the maximal cliques or the maximum clique of a graph. If the analysis of the models is done online, efficiency considerations should be taken into account.

In this paper we will consider that the structure mining step is done starting from the complete set of structures learned by the EDA in one or more runs. We present results in the identification of frequent subgraphs in the structures of the models learned by EDAs and the relationships of these subgraphs with the original structure of the problem.

3.3 Visualization of the relevant features

Visualization techniques are an important tool for interpretation and understanding of data. Classical approaches dealing with visualization of small, isolated problems have recently shifted to the visualization of massive scale, dynamic data comprised of elements of varying levels of certainty and abstraction [25].

Now we consider the characteristics of the data generated in EDAs. We use the following classification to distinguish the way the different classes of data are produced:

- Data related to the points generated in relation to the search space (e.g. number and distribution of the points generated).
- Data related to the evaluation of the (possible multiple) objective functions (e.g. quality of the solutions, shape of the Pareto front approximations, etc.).
- Probabilistic models (e.g. structure and parameters of the probabilistic models).

Notice that in opposition to other types of data derived from physical processes (e.g. ecological and biological systems), EDAs, and in general evolutionary algorithms, are algorithmic simulations. All the components of the algorithm are well specified and the obtained data is produced by the process itself. Therefore, the conditions of the simulations can be modified to study its influence in the data obtained.

We will focus on the application of visualization techniques to unveiling different features of the probabilistic learned by EDAs. In the following we describe a number of preprocessing and visualization techniques that will be used in our experiments.

Among the visualization techniques used for multi-dimensional categorical data are *parallel coordinates*, *dendrograms* and *glyphs*.

In parallel coordinates [11], every observation is plotted for each axis/variable, and a connecting line is drawn for each observation between all the axes. This technique can be used to identify outlier points, those that do not follow the same trend than the rest. Dendrograms are graphs that serves to represent hierarchical trees [12]. A dendrogram consists of many U-shaped lines connecting objects in the hierarchical tree. The height of each U represents the distance between the two objects being connected. A glyph is a visual representation of a piece of data where the attributes of a graphical entity (e.g. shape, size, color, and position) are dictated by one or more attributes of a data record. The placement or layout of glyphs on a display can communicate significant information regarding the data values themselves as well as relationships between data points [26].

Frequency matrices [4, 8, 24] are those where the frequency in which each arc appears in the Bayesian network is represented. The frequencies of the two arcs that involve the same pair of variables are counted together. Frequency matrices can be computed for a particular generation or taking the information from all the generations.

The frequency matrix representation has two main limitations. First, it is not possible to investigate the types of structures learned at each generation. It has been early observed that structures of the models learned by EDAs can change along the evolution [2]. The other limitation of this type of representation is that it is not possible to capture interactions between different substructures of the problem. For instance, if the frequency matrix shows that two edges have occurred ten times in the structures learned, we cannot determine how often they have appeared together in the same structure.

4. PREVIOUS WORK

Our work is part of an ongoing research trend that uses the analysis of the structures of the probabilistic models learned by EDAs to investigate their relationship with the original problem structure. A number of researchers have studied the most frequent dependencies learned by the probabilistic models in EDAs and analyzed their mapping with the function structure [2, 17, 18, 24].

More recently, some work has been devoted to analyzing the way in which the different components of the EDA influence the arousal of dependencies [8] and to use the probabilistic models obtained by EDAs to speed up the solution of similar problems in the future [9]. Recently, it has been shown [16] that the model quality in EDAs based on Bayesian networks can be greatly influenced by the scoring metric employed and the type of selection operator (and its intensity).

Perhaps the most common visualization technique used to show the structural relationships in the model learned by EDAs are frequency matrices [4, 8, 24] which are usually represented using choropleth maps (maps with color-shadings to represent quantities) or contour plots.

Other visualization methods commonly used in EDAs are dotplots, histograms, boxplots and scatter plots.

There is an extensive list of papers treating the application of EDAs to discrete and continuous multi-objective problems. The review of this work is beyond the scope of

this paper. Relevant to our research is the application of Bayesian network-based EDAs to discrete problems as done in [13, 15, 22]. This work is focused on the solution of the optimization problem and does not present a detailed analysis of the structures of the Bayesian networks learned by the EDAs. The type of multi-objective problems used in our experiments has been previously used to study EDAs in [3].

5. EXPERIMENTS

In this section, we investigate the arousal of dependencies in EBNA for multi-objective problems by analyzing the relationship between the structures learned by EBNA and the structure of the original problem. The main objective is to determine whether original interactions of the problem are translated in the models learned and in which way these original structural relationships are reflected in the learned models. We emphasize the role of the techniques for extracting, processing and visualizing the information from the models.

5.1 Multi-objective SAT problem

As a benchmark for our experiments we use a multi-objective satisfiability (SAT) problem.

Let $\mathbf{U} = \{U_1, U_2 \cdots U_n\}$ be a set of n Boolean variables. A (partial) truth assignment for $\mathbf{U}$ is a (partial) function $T : \mathbf{U} \rightarrow \{true, false\}$. Corresponding to each variable U_i are two literals, u_i and $\neg u_i$. A literal u_i (resp. $\neg u_i$) is $true$ under T iff $T(u_i) = true$ (resp. $T(u_i) = false$). A set of literals is called a clause, and a set or sequence (tuple) of clauses a formula ϕ. The satisfiability problem (SAT) is the problem of finding a satisfying assignment for a formula.

We address the Max-SAT problem which consists of finding an assignment of the literals that maximizes the number of satisfied clauses. We focus on SAT problems where each clause has exactly three literals (3-SAT). Besides, the single objective problem is extended to the multi-objective domain by considering a set of k formulas to be simultaneously satisfied.

Let $\{\phi_1, \ldots \phi_k\}$ be a set of formulas, we define the set of functions $\{f_1, \ldots f_k\}$, $f_i(\mathbf{x}) \rightarrow \mathcal{N}$, where $f_i(\mathbf{x})$ gives the number of satisfied clauses in ϕ_i for the unique assignment of the literals determined by $\mathbf{x}$. $\mathbf{x} \in \mathcal{X} = \{0, 1\}^n$ and $x_i = 1 \rightarrow U_i = u_i, x_i = 0 \rightarrow U_i = \neg u_i$.

5.1.1 Function benchmark

We generate different classes of multi-objective satisfiability problems trying to mimic different structural relationships between the variables and formulas. Four classes of problems are used. For each class, 10 instances of a multi-objective 3-SAT problem are generated. Each problem has 20 variables and 10 formulas, with 20 clauses each one. The classes were generated as follows:

- Class 1: For each formula, clauses are independently and randomly generated.
- Class 2: Clauses are assigned different probabilities to be included in a formula. The probability is higher if the clause contains variables from a predefined set Z. Formulas are independently generated.
- Class 3: Half of the formulas are generated as described for class 2. Then, the rest of formulas are generated by modifying previous formulas in the following way: First a copy ϕ_j of formula ϕ_i is made, then a number of variables are replaced by variables that are not in the current formula.
- Class 4: Half of the formulas are generated as described for class 2. Then, the rest of formulas are generated by modifying previous formulas in the following way: First a copy ϕ_j of formula ϕ_i is made, then a random number literals are negated.

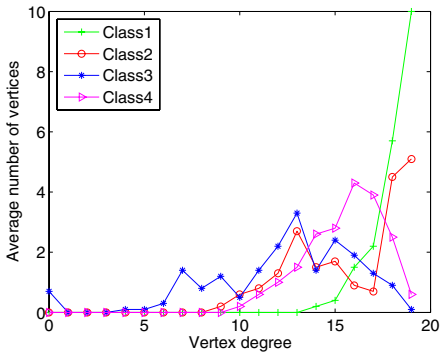

Figure 1: Average number of vertices for each vertex degree in the interaction graphs.

As an initial step to understand the characteristics of the instances, we compute the interaction graphs corresponding to each instance. In these graphs, two variables are joined by an edge if they interact in the problem structure. For the multi-objective SAT problem, we consider as interacting variables those that appear together in at least one of the clauses of any of the formulas. From these graphs, we compute different statistics that can provide information about particular characteristics of the problem. Figure 1 shows the average vertex degree distribution in the four classes of instances. It can be seen that there are important differences between the instances in the distribution of vertices with different degrees. While for class 1, half of the total number of vertices are connected to all the other vertices, for class 3 the number of vertices with high degree is much smaller. We expect these differences to have an influence in the structures learned by the EDA.

5.2 Design of the experiments

We use EBNA as described by Algorithm 1 to find Pareto set approximations for the 40 problem instances. The algorithm used a population size of 500 individuals and it was allowed a maximum number of 50 generations. 30 independent executions where conducted for each instance. In general, the Pareto set approximations achieved by EBNA were better (using different metrics of comparison) than those achieved for the same problems with EDAs that used simpler probabilistic models (data not shown).

The following analysis focuses on the Bayesian network structures generated. The total number of such structures was 60000.

5.3 Numerical results

We start from a set of initial experiments that show some topological properties of the models learned by the algo-

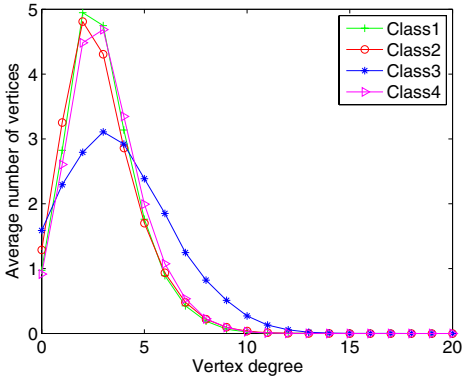

Figure 2: Average number of vertices for each vertex degree in the learned structures.

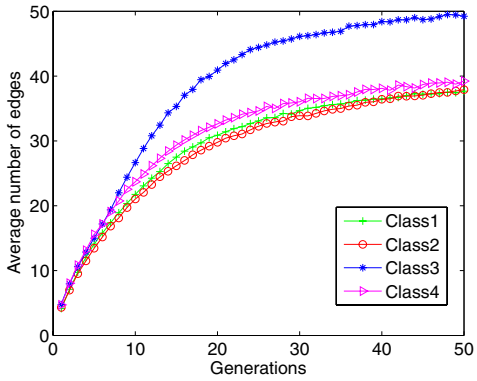

Figure 3: Average number of edges learned at each generation.

rithm. Then, we investigate the relationship between the original structure of the problem (the clauses in the original formulas) and the most frequent clusters of three variables found in the Bayesian networks learned by EBNA. Finally, some of the visualization techniques are used to capture more insight about the structural relationships between the components of the Bayesian networks learned.

5.3.1 *Complexity of the learned structures*

In the first experiment the degree distribution from the Bayesian learned structures is computed. For each instance, we use the 1500 available structures to compute the corresponding mean degree distribution. The obtained statistics are used to find the degree distribution corresponding to each class of problems. The results are shown in Figure 2.

It can be seen in Figure 2 that the distribution is similar for classes 1,2 and 4. However, the degree distribution of instances in class 3 is different, there are less vertices with low distribution and more vertices with higher distribution. Notice, that this is a behavior different from what expected from considering the original problem degree distribution shown in Figure 1. This could be explained by three reasons: 1) The EDAs start from a uniform population and initial structures contain few edges, i.e. the degree of all vertices is very low. 2) The fact that two variables appear together in a clause does not necessarily imply that these variables have a high interaction. As a result the number of edges in the probabilistic model, and the degree of the vertices, can be smaller than in the original interaction graph. 3) The learning algorithm imposes constraints on the maximum number of parents for each variable in the Bayesian network. Therefore, some of the original interactions may not be represented.

The analysis of the degree distribution hints to the fact that the structures learned for problems of class 3 are more dense than for the rest of the problem. We analyze the average number of edges learned for each class of problem at each generation. These results are shown in Figure 3. The number of dependencies learned increases over time. We hypothesize that the increasing level of diversity enforced by the application of the Pareto rank selection method induces more complex models.

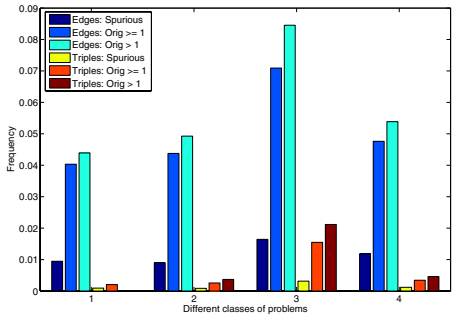

Figure 4: Frequency of different sets of edges in the learned structures.

5.3.2 *Relationship with the original problem structure*

We start by evaluating to which extent the original structure of the problem is translated to the Bayesian networks. We hypothesize that the sets of three variables that belong to a clause in any of the original formulas will appear together with higher frequency in the structures learned by the models. Similarly, we expect this fact to occur for pairs of variables that belong to the same clause. Therefore, for each of the 40 instances we compute the frequency of appearance of each possible pair and triple of vertices in the corresponding Bayesian networks. Notice that this implies computing of possible cliques of size 3 in each of the networks. There are 210 possible pairs and 1140 triples. Then, we compute the statistics for pairs that were in the original structure and those that were not. These numbers were averaged for each of the problem classes and the results are shown in Figure 4.

In Figure 4, "Edges: Spurious" means the pair of vertices that were not together in any of the clauses of the original problem. "Edges: Orig>=1" means edges that were in at least one of the clauses. "Edges: Orig>1" means edges that were in at least two of the original clauses. For triples the interpretation of the legends is respectively the same.

The main conclusion from the analysis of Figure 4 is that

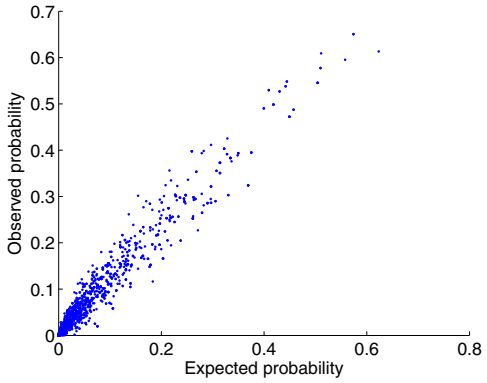

Figure 5: Mapping between the expected and observed probabilities of appearing in a learned structure for each original clause of all instances of problem class 3.

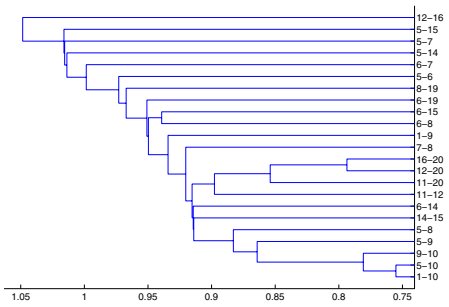

Figure 6: Dendrogram visualization of the selected edges of instance $S_{3,2}$.

the pairs of vertices and triples in the learned structures are more likely to be related in the original structure of the problem. Also, if pairs and triples of variables appear more often in the clauses they will have a higher probability to be captured in the model. However, not all of the original interactions have to be captured by the model. It can also be observed that there are differences in the frequency of the structures between the three classes of problems. In particular, there are more pairs and triples for instances of class 3. This is consistent with the experiments described before.

We investigate now whether the frequency of triples can be explained by the joint effect of the pairs of variables that are comprised in each triple. This is, for every triple (x_a, x_b, x_c) corresponding to a clause in the original problem, we measure the correlation between the expected probability of the triple $p_e(x_a, x_b, x_c) = p_o(x_a, x_b) \cdot p_o(x_a, x_c) \cdot p_o(x_b, x_c)$ and the observed probability $p_o(x_a, x_b, x_c)$. This computation is done for all the triples in all the problem instances. The probability values corresponding to the pairs and triples are computed as explained in the previous experiment.

Figure 5 shows the mapping between $p_e(x_a, x_b, x_c)$ and $p_o(x_a, x_b, x_c)$ for all instances of problem class 3. The average correlation values are above 0.85 for all the instances. This indicates that the univariate product of edges frequency can be used to predict the appearance of the triples.

5.3.3 *Application of the visualization techniques*

The process in which small structural components (e.g. the edges) are combined to form bigger structures in the models is of particular interest. We have seen in the previous section that the appearance of a triple can be predicted by looking at the frequencies of the edges that integrate it. However, we would like to know the implication of this aggregation effect for the emergence of larger structures and the common appearance of groups of related triples.

It is difficult to detect how often different components appear together forming increasingly complex substructures. A statistical analysis of this kind is hard to carry out and even for a single problem there arise obstacles to accurately detect relationships between the structural components.

We use hierarchical clustering and dendrogram visualization to show the way in which different components of the problem are related in the structures learned. For one of the instances (second instance of class 3, namely $S_{3,2}$ instance), we mine the 1500 structures corresponding to the 30 executions of EBNA for this problem. First, the triples with a frequency of appearance above 0.25 are extracted and the edges that form these triples are identified. There are 23 edges. For each of the 1500 structures, the substructures comprising the 23 edges are extracted, i.e. for each structure, a binary vector indicating whether each edge was learned in the corresponding Bayesian network is produced.

Finally, the 1500 substructures are clustered using hierarchical clustering and taking as the distance measure the opposite of the correlation between the observations, i.e. edges that appear together in the structures with a high frequency are closer. The hierarchical cluster obtained in this way is shown in Figure 6. This graph gives an idea of the proximity of the edges in terms of their frequency of appearing together in the structures. Edges joining pair of variables that belong to the same clause in the original problem tend to appear together more frequently. More complex structures appear, although less frequently. The important gain is that we can identify how these more complex subgraphs are formed. The same procedure can be used in other problems to display the hierarchical organization of the interactions between the problem components.

One limitation of the dendrogram visualization is that it is not possible to observe the time (generation) at which two or more edges appear together in the structures. We use parallel coordinate visualization to reveal this information. Using the same data set of substructures learned for instance $S_{3,2}$ and adding the generation at which each structure was learned, we generate the parallel coordinates graph shown in Figure 7 where edges are represented using their index in an arbitrary ordering.

In this graph, there are only two possible types of lines between two adjacent points representing different edges. If there is a horizontal line between (x, g) and $(x + 1, g)$, this means that the edges x and $x + 1$ appear together in the Bayesian network learned at generation g in at least one of the 30 possible runs. If there is a line from $(x, 0)$ to $(x+1, g)$ it means that in at least one of the structures, edge x appears in the Bayesian network learned at generation g but edge $x + 1$ is not present in the same structure.

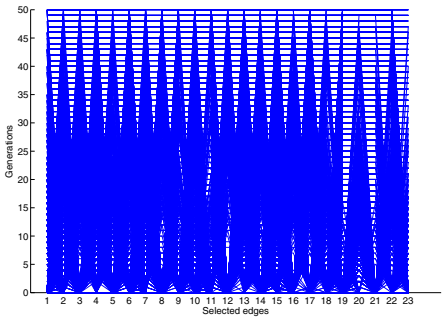

Figure 7: Parallel coordinate visualization of the selected edges of instance $S_{3,2}$.

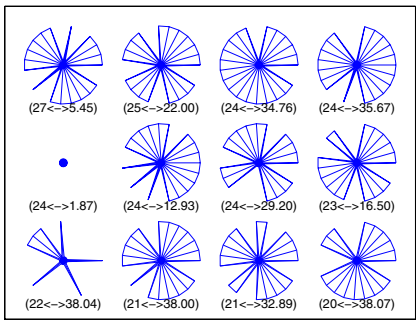

Figure 8: Glyph visualization of a number of exemplars for the selected edges of instance $S_{3,2}$.

It can be observed that the frequent cone patterns indicate that it is common to find, in the learned structures, only one of the two edges represented by the adjacent points. However, this pattern is not shared by edge 19 and its adjacent edge 20. In this case, the two edges seem to appear always together (or be both absent) in the structures, at least this can be clearly observed for generations above $g = 15$. For $g < 15$, cluttering does not allow a clear visualization. The relationship between these edges is the type of outliers that can be detected with this type of techniques.

When the number of structures generated is too high it is important the identification of a set of representative structures. We have applied a clustering algorithm that separates the data into non-overlapping clusters and simultaneously identify a characteristic member of the cluster or exemplar. Affinity propagation [6] is applied to the set of 1500 substructures obtained for instance $S_{3,2}$.

The algorithm grouped the 1500 structures into 124 clusters. We then computed the average generation for structures that belong to the same cluster. The idea was to determine if the clustering was able to detect some common pattern in the structures that could be associated to the generation in which they were learned.

Figure 8 shows the star glyph representation of each exemplar for the 12 clusters with more members. A star glyph represents each observation as a "star" whose i-th spoke is proportional in length to the i-th coordinate of that observation. Each star has 27 spokes, one for each edge, and observations can only be one or zero respectively meaning that the corresponding edge is present or absent in the structure of the exemplar. At the bottom of each glyph, two values are shown: The first one corresponds to the number of structures in the cluster, the second is the average generation in which the structures in the cluster were learned.

The analysis of the glyphs provides information about the similarities between the clusters. For example, it can be appreciated that one empty structure is identified as the exemplar of a cluster that comprises 24 structures. Not surprisingly, the average generation in which the structures that belong to this cluster were learned is very low (1.87). This fact corresponds to the evidence that at the first generations of the EDA the Bayesian networks structures contain very few edges. Notice that the clustering algorithm does not employ information about the generations at which structures were learned. In this and other similar problems, glyphs constructed from the exemplars can help to reveal more sophisticated patterns in the structures.

6. CONCLUSIONS

EDAs have shown their capacity to solve optimization problems. Still they are very useful at revealing details of the problem structure. In order to enhance this capacity it is important: 1) to develop the tools to extract and visualize the information from the models, 2) to understand the way in which the EDA components influence the arousal of dependencies, 3) to expand the uses of the structural information captured by the models in the conception of more advanced (hybrid) optimization schemes.

In this paper we have investigated topics related to the first two research lines mentioned above. We have shown the way in which different data analysis and visualization techniques can help to discover the structural relationships in the models learned by EDAs. Although these techniques are widely applied in other domains, they have not been employed to investigate the data sets of graphical structures generated by EDAs. Indeed, in this domain they are required to unveil meaningful information.

We have partially addressed the relationship between the original interactions of multi-objective problems and the dependencies that are captured by the Bayesian networks learned in the evolution of EDAs. For the multi-objective SAT problems considered in this paper, we have shown that variables that appear together in the formulas are more likely to be related in the Bayesian networks. Nevertheless, the influence of the different objectives in the structures learned by the models, and other factors related to multi-objective nature of the problem have been not investigated. These are relevant topics worth of future research.

7. ACKNOWLEDGMENTS

This work has been partially supported by the Etortek, Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government), TIN2008-06815-C02-01, TIN-2008-06815-C02-02, TIN2007-62626 and Consolider Ingenio 2010 - CSD2007-00018 projects (Spanish Ministry of Science and Innovation) and COMBIOMED network in computational biomedicine (Carlos III Health Institute).

8. REFERENCES

[1] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences (PNAS)*, 97(21):11149–11152, 2000.

[2] E. Bengoetxea. *Inexact Graph Matching Using Estimation of Distribution Algorithms*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2003.

[3] P. A. Bosman. *Design and Application of Iterated Density-Estimation Evolutionary Algorithms*. PhD thesis, Universiteit Utrecht, Utrecht, The Netherlands, 2003.

[4] C. Echegoyen, R. Santana, J. A. Lozano, and P. Larrañaga. The impact of probabilistic learning algorithms in EDAs based on Bayesian networks. In *Linkage in Evolutionary Computation*, Studies in Computational Intelligence, pages 109–139. Springer, 2008.

[5] R. Etxeberria and P. Larrañaga. Global optimization using Bayesian networks. In A. Ochoa, M. R. Soto, and R. Santana, editors, *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 151–173, Havana, Cuba, 1999.

[6] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

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

[8] M. Hauschild, M. Pelikan, C. Lima, and K. Sastry. Analyzing probabilistic models in hierarchical BOA on traps and spin glasses. In D. Thierens et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2007*, volume I, pages 523–530, London, UK, 2007. ACM Press.

[9] M. Hauschild, M. Pelikan, K. Sastry, and D. E. Goldberg. Using previous models to bias structural learning in the hierarchical BOA. MEDAL Report No. 2008003, Missouri Estimation of Distribution Algorithms Laboratory (MEDAL), 2008.

[10] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[11] A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and its Applications*. Springer, 2007.

[12] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 2:241–254, 1967.

[13] N. Khan, D. E. Goldberg, and M. Pelikan. Multi-objective Bayesian optimization algorithm. IlliGAL Report No. 2002009, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, 2002.

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

[15] M. Laumanns and J. Ocenasek. Bayesian optimization algorithm for multi-objective optimization. In J. J. Merelo, P. Adamidis, H. G. Beyer, J. L. Fernández-Villacañas, and H. P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VII*, volume 2439 of *Lecture Notes in Computer Science*, pages 298–307, Granada, Spain, 2002. Springer Verlag.

[16] C. F. Lima, F. G. Lobo, and M. Pelikan. From mating pool distributions to model overfitting. In M. Keijzer, editor, *Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation*, pages 431–438, New York, NY, USA, 2008. ACM.

[17] 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 *Proceedings of the 2007 Congress on Evolutionary Computation CEC-2007*, pages 1083–1090. IEEE Press, 2007.

[18] H. Mühlenbein and R. Höns. The estimation of distributions and the minimum relative entropy principle. *Evolutionary Computation*, 13(1):1–27, 2005.

[19] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV*, volume 1141 of *Lectures Notes in Computer Science*, pages 178–187, Berlin, 1996. Springer Verlag.

[20] K. Murphy. The BayesNet toolbox for Matlab. *Computer science and Statistics: Proceedings of Interface*, 33, 2001.

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

[22] M. Pelikan, K. Sastry, and D. E. Goldberg. Multiobjective hBOA, clustering and scalability. IlliGAL Report No. 2005005, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL, February 2005.

[23] R. Santana, C. Echegoyen, A. Mendiburu, C. Bielza, J. A. Lozano, P. Larrañaga, R. Armañanzas, and S. Shakya. MATEDA: A suite of EDA programs in Matlab. Technical Report EHU-KZAA-I2/09, Department of Computer Science and Artificial Intelligence, University of the Basque Country, February 2009.

[24] R. Santana, P. Larrañaga, and J. A. Lozano. The role of a priori information in the minimization of contact potentials by means of estimation of distribution algorithms. In *Proceedings of the Fifth European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, volume 4447 of *Lecture Notes in Computer Science*, pages 247–257. Springer Verlag, 2007.

[25] J. J. Thomas and K. A. Cook. A visual analytics agenda. *IEEE Computer Graphics and Applications*, 26(1):10–13, 2006.

[26] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210, 2002.