10th Brazilian Congress on Computational Intelligence (CBIC'2011), November 8 to 11, 2011, Fortaleza, Ceará Brazil

Score Metrics for Learning Bayesian Networks used as Fitness Function in a Genetic Algorithm

Edimilson B. dos Santos[1], Estevam R. Hruschka Jr.[2] and Nelson F. F. Ebecken[1]

[1]COPPE/UFRJ - Federal University of Rio de Janeiro.
Rio de Janeiro, Brazil.
[2]DC/UFSCar – Federal University of São Carlos.
São Carlos, Brazil.
edimilson.batista@coc.ufrj.br
estevam@dc.ufscar.br
nelson@ntt.ufrj.br

Abstract – Variable Ordering (VO) information may be used as a constraint to reduce the search space of learning Bayesian Networks (BNs) from data. Several authors have proposed the use of Evolutionary Algorithms (EAs) to find such an ordering. In this work, a genetic algorithm named VOGA (Variable Ordering Genetic Algorithm) has been applied to this goal. Since the fitness function plays an important role in the performance of genetic algorithms, we present and discuss five score metrics, applied to induce BNs from data, to be used as fitness function in VOGA. The main objective is to investigate the VOGA performance when employing different metrics.

1 Introduction

Bayesian Networks (BNs) are graphical representations of probability distribution and have been widely used to representation of uncertainty in Artificial Intelligence. According to [15], BNs are very important in modern expert systems, diagnostic machines and decision support systems.

Good results reported in the literature over the last years have motivated the development of many BN learning algorithms [16]. The development of automatic methods for learning the network directly from data is relevant problem. It is well known that the search space for a BN with n variables has exponential dimension. Therefore, finding the BN structure that represents the dependencies among the variables is a hard task, considered a NP-Complete problem [8]. As a consequence, methods that induce approximate models by reducing the search space (imposing some restrictions) have been developed. A very common restriction when learning a BN is to require information about what is called Variable Ordering (VO) (a list of variables ordered by the relevance they play in the process) [19]. The same situation happens when trying to learn a Bayesian Classifier (BC) from data. However, to find such an ordering for the variables in a BN is a complex problem which requires a lot of information about the model [5].

Several authors have proposed the use of Evolutionary Algorithms (EAs) [2] to find a suitable VO in order to optimize the learning of BN structures. EAs have characteristics that make them very efficient as search and optimization tools for most different types of problems [12]. In [20] the authors have proposed a genetic algorithm (which is an EA) named VOGA (Variable Ordering Genetic Algorithm) which identify a suitable VO for inducing BN structures. VOGA has used a Bayesian score (g function), defined in K2 algorithm [9], as its fitness function. The score metric to compute the joint probability of a Bayesian network structure and a given database plays a crucial role in K2-like Bayesian methods [23].

Researchers have proposed a variety of score metrics based on different assumptions to induce a BN from data. The score metric that performs best is of interest [23]. Since VOGA has used a score metric as fitness function, we present, in this paper, other four score metrics that can also be used as fitness function for VOGA: Entropy [4], BDe (Bayesian Dirichlet equivalent) [14], MDL (Minimum Description Length) [3] and AIC (Akaike Information Criterion) [1]. Our main goal is to assess the performance of VOGA with different score metrics and verify which ones (or if a combination of them) may improve it.

The remainder of this paper is organized as follows. In Section 2 an overview of the foundations of BNs, along with the structure learning problem, the score metrics, and the fundamentals of EAs are briefly presented. Section 3 introduces some related works found in literature. In the Section 4, the performed experiments are described and discussed. Finally, Section 5 brings the concluding remarks and points out some future work.

2 Basic Concepts and Procedures

2.1 Bayesian Networks

Bayesian Networks (BNs), also known as belief networks or causal networks, are graphics models which the nodes represent the random variables and the arcs represent the direct connection between them. A BN consists of two major components: i) a directed acyclic graph (DAG) – or graphical structure – and ii) a conditional probability table (CPT) – or

numerical parameters.

The DAG $G = (V, E)$ consists of a nodes set ($V = \{X_1, X_2, ..., X_n\}$) and an ordered pairs set (E) of distinct elements in V. Each node in V corresponds to a discrete random variable in the domain. The elements of E are named edges (or arcs). An edge, $X_i \rightarrow X_j$ describes a parent and child relation, where X_i is the parent and X_j is the child. All parents of X_j constitute the parent set of X_j, which is denoted by Π_{Xj}. In addition to the graph, each node X has an associated CPT specifying the probability of each possible state of X given each possible combination of states of the nodes in Π_X. If a node X is such that $\Pi_X = \varnothing$, its associated table gives the marginal probabilities of X.

In the process of learning BNs from data the BN variables represent the dataset attributes and, for this reason, the words 'variable', 'attribute' and sometimes 'node' will be interchangeably used in this paper. When using algorithms based on heuristic search, the initial order of the dataset attributes is a relevant issue since it may contribute to a better (or worse) result. Some of the algorithms use the variable ordering (VO) to determine the arc directions such that a variable is a possible parent only to those variables that follow it in the list. The VO information may be used as a constraint to reduce the search space, as done by the K2 algorithm [9].

2.2 The K2 Algorithm

The K2 algorithm [9] constructs a BN from data using a heuristic search procedure. It receives as input a complete dataset and a VO. K2 is broadly used due to its good performance in terms of computational complexity (time) as well as its good results when an adequate VO is supplied [17].

The algorithm uses a greedy process to search for the best structure. Given a dataset D with M objects and a VO, it begins as if every node has no parent. Then, beginning with the second attribute from the ordered list (the first one is the root node), its eligible parents are evaluated and those that maximize the whole probability structure are added to the network. The process is repeated for all attributes in order to get the best possible structure. The K2 metric for evaluating each possible parent set to each variable is defined by equation (1):

$$g(x_i, \pi_{x_i}) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \qquad (1)$$

where each attribute x_i ($i = 1,\ldots,n$) has r_i possible values $\{v_{i1}, v_{i2}, \ldots, v_{ir_i}\}$. D is a dataset with m objects. Each attribute x_i has as its set of parents π_{x_i}, and q_i is the number of instantiations of π_{xi}. N_{ijk} is the number of objects in D, in which x_i has value v_{ik} and π_{x_i} is instantiated as w_{ij} (w_{ij} represents the j-th instantiation relative to D of π_{x_i}). Finally, $N_{ij} = \Sigma N_{ijk}$. The g function of each variable is added to compose the score of the network.

Having defined the best structure, the network conditional probabilities are determined using a Bayesian estimation of the (predefined) network structure probability.

2.3 Score Metric

The Bayesian approach requires a quality measurement which demonstrates the suitability of the network structure being a proper representation for a given dataset. Besides the g function, presented in subsection 2.2, other four measures are presented below.

1) Entropy: Entropy is a non-negative measure of uncertainty which is maximal when uncertainty is maximal and zero when there is complete knowledge; the more information is given the lower the entropy [3].

Let the entropy metric $H(B_s, D)$ of a network structure and database be defined as [4]:

$$H(B_s, D) = -N \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N_{ij}} \qquad (2)$$

and the number of parameters K as

$$K = \sum_{i=1}^{n} (r_i - 1) \cdot q_i \qquad (3)$$

2) Akaike Information Criterion – AIC: Akaike [1] introduced the information criterion for model selection, generalizing his earlier work on time series analysis and factor analysis.

The AIC metric $Q_{AIC}(B_S, D)$ of a Bayesian network structure B_S for a database D is:

$$Q_{AIC}(B_S, D) = H(B_S, D) + K \qquad (4)$$

3) The Bayesian Measure: The basic idea of the Bayesian approach is to maximize the probability of the network structure given the data, that is, to maximize $P(B_S|D)$ over all possible network structures B_S given the cases of the database D. To this end, the probability given the database is calculated for various network structures and the one with the highest probability is selected [3].

The Bayesian metric of a Bayesian network structure B_D for a database D is:

$$Q_{Bayes}(B_S, D) = P(B_S)\prod_{i=0}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij}+N_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(N'_{ijk}+N_{ijk})}{\Gamma(N'_{ijk})} \quad (5)$$

where $P(B_S)$ is the prior on the network structure and $\Gamma(.)$ the gamma-function. N'_{ij} and N'_{ijk} represent choices of priors on counts restricted by $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk}$. With $N'_{ijk}=1$ (and thus $N'_{ij}=r_i$), we obtain the K2 metric (function (1)). With $N'_{ijk}=1/r_i.q_i$ (and thus $N'_{ij}=1/q_i$), we obtain the BDe metric (Bayesian Dirichlet equivalent) [4].

4) Minimum Description Length – MDL: The MDL principle stems from coding theory where the aim is to find an as short as possible description of a database with as few parameters as possible. The MDL measure can be regarded as an approximation of the Bayesian measure, and thus has a Bayesian interpretation. However, it offers several advantages over the Bayesian measure [3].

The MDL metric $Q_{MDL}(B_S,D)$ of a Bayesian network structure B_S for a database D is defined as:

$$Q_{MDL}(B_S, D) = H(B_S, D) + \frac{K}{2}\log N \quad (6)$$

2.4 Evolutionary Algorithms

Some heuristics have been proposed to find a suitable VO, since the VO information may not be available in real world applications [6]. In this paper, an EA using different feature ranking metrics to create new fitness functions is proposed.

Evolutionary Algorithms (EAs) [2] are computational models that solve a given problem by maintaining a changing population of chromosomes (individuals), each with its own level of "fitness". A fitness function is used to measure the quality of each chromosome (to select those that better fit the problem). Such algorithms vary according various models of evolutionary computation which can be divided into three main categories: Genetic Algorithms, Evolution Strategies and Genetic Programming [2]. The main differences between them are in the representation of individuals and, consequently, the type of genetic operators used. There are other differences, such as the order in which they performed some of the operations and the selection methods used and more details on such differences can be found in [2].

A Genetic Algorithm (GA) can be seen as a class of EA having two classic operators, namely crossover and mutation. The crossover operator tries to get new chromosomes that inherit good characteristics of its parents, thus improving the quality of the population. The mutation operator tries to prevent local minimum, searching other areas of search. The stop condition may be determined using a fixed number of generations or a fitness threshold.

3 Related Works

Scoring methods use a score to evaluate the consistency of the current structure with the probability distribution that generated the data. A good score metric should be able to identify the true network structure from several candidates, if the true one is among the candidates. Most of the score metrics are proposed independently and are quite successful in some applications. Obviously, the performance of a learning algorithm depends not only on the score metric but also on the search method.

In [23], the authors present a comparison of five types of score metrics for Bayesian network learning: 1) UPSM (Uniform Prior Score Metric); 2) CUPSM (Conditional Uniform Prior Score Metric); 3) DPSM (General Dirichlet Prior Score Metric); 4) likelihood-equivalence Bayesian Dirichlet score metric (BDe); and 5) MDL score metric. They use the K2-like greedy search method throughout all experiments and apply different score metrics to structure learning. The authors have conducted several experiments to identify the true network structure from possible candidates and have compared the performances of the five types of score metrics. The experimental results have shown that the UPSM, the CUPSM, the DPSM, and the MDL are able to correctly identify the true network structure, while the BDe may fail to identify the true network structure if its equivalent sample size and distribution orders are not chosen properly.

In the literature, we find evolutionary methods applied to the training of a Bayesian network structure using different score metrics. Initial works are presented in [11][18]. In this works, the structure is build using a genetic algorithm and with or without the knowledge of a topologically correct order on the variables of the network. In [18] an evolutionary algorithm is used to conduct research over all topologic orders and then the K2 algorithm is used to train the model.

In [21] structures are learned using the MDL criterion. The algorithm, named MDLEP, does not require a crossover operator but is based on a succession of mutation operators. An advanced version of MDLEP named HEP (Hybrid Evolutionary Programming) was proposed in [22].

Chen *et al* propose a new method for computing the *log(P(S,D))*, which is used to evaluate the fitness of the network structure in their work [7]. The simulation shows the validity and efficiency of the proposed approach.

In [20] a hybrid method has been proposed. It combines the theory of EAs and BNs. This method named VOGA (Variable Ordering Genetic Algorithm) aims at optimizing the learning of network structure by searching a suitable VO. The method uses a genetic algorithm which represents possible variable orderings as chromosomes. The identification of variables (ID) is codified as an integer number. Therefore, each chromosome has (n -1) genes, where n is the number of variables (including the class variable) and each gene is instanced with a variable ID. Thus, each possible ordering may form a chromosome. The fitness function used to assess each chromosome is given by the Bayesian score (g function), defined in K2 algorithm. Fig 1 presents a flowchart that summarizes the process all.

In this paper, we want to verify if the performance of VOGA may be improved by applying other score metrics. Therefore, we intend to incorporate new score metrics in VOGA and analyze what influences they cause in the search of the suitable VO.

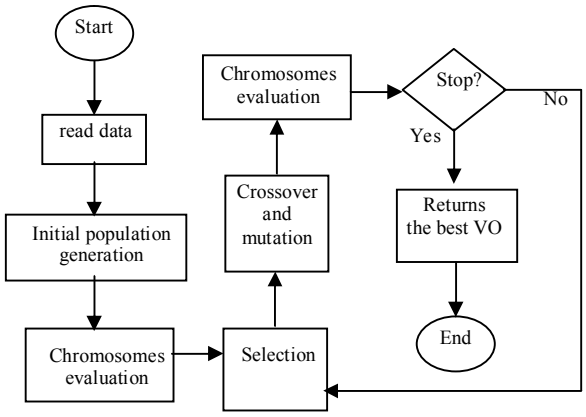

Figure 1 – VOGA flowchart.

4 Experiments and Analysis of Results

In this paper, new VOGA [20] versions are proposed, each one having a different score metric as fitness function, namely g function, MDL, AIC, BDe and Entropy. Besides these measures, the combination them is performed in two ways to create two new fitness functions: i) the sum of all these score metrics is used to evaluate each chromosome; ii) the values of all these metrics are used as votes to evaluate each chromosome and the majority voting strategy is applied.

The experiments performed with VOGA using different score metrics have involved six knowledge domains which have its characteristics summarized in Table 1. The Synthetic 1, Synthetic 2 and Synthetic 3 domains have 32 variables each and were created using a sampling strategy applied to the structures of previously known Bayesian Networks described in Figure 2. The Synthetic 50-50, Synthetic 50-100 and Synthetic 50-200 domains have 50 variables each and different number of arcs each (50 arcs, 100 arcs and 200 arcs respectively). The structures of those three BNs are shown in Figure 3. The different numbers of arcs were generated to verify the impact of the network density in the task of obtaining optimal orderings.

Table 1 – Dataset description with dataset name (Domain), number of attributes plus class (AT), number of instances (IN) and number of classes (CL).

	Synth 1	**Synth 2**	**Synth 3**	**Synth 50-50**	**Synth 50-100**	**Synth 50-200**
AT	32	32	32	50	50	50
IN	10000	10000	10000	50000	50000	50000
CL	2	2	2	2	2	2

The following steps define the experimental methodology for each of the six datasets in more detail:

1) The experiments involved VOGA using crossover (OX) and mutation (based on order) canonical operators, as described in [20].

2) In VOGA, the crossover rate has been set to 0.8 and the mutation rate has been initialized with 0.3. These values have been defined empirically.

3) For a given dataset, all the algorithms were assessed in a number of runs, using the same initial population.

4) Each dataset was run by VOGA versions using the five scores metrics (g function, Entropy, MDL, AIC and BDe). Table 2 presents a comparison among the network structures induced by VOGA and the original ones through number of extra

arcs, number of missing arcs and number of reverse arcs.

5) Two other VOGA versions have been developed using a combination of the five aforementioned metrics as fitness function. The first combination strategy evaluates the chromosomes through the majority voting. That is, each chromosome (ordering) is assessed by the five score metrics. The chromosome having the highest values of scores (for each metric) will have more votes and will be considered the best choice. This algorithm is named VOGA – "Majority vote". The second version uses the sum of all score as fitness function and is named VOGA – "Sum of all scores". Table 2 also shows the differences between the original BNs and induced ones obtained by these versions.

6) All executions end after 10 generations without improvement and return the best ordering found so far, as well as the corresponding learned BN. Table 3 presents the number of generations for convergence.

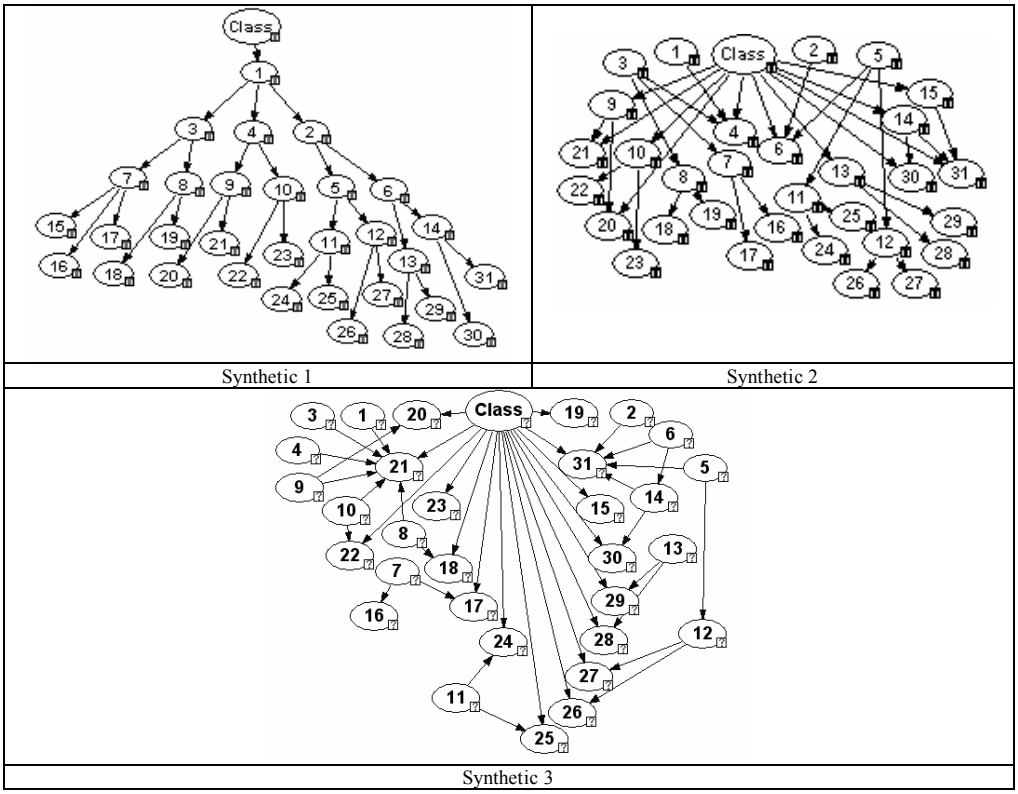

Figure 2 – Bayesian Networks representing Synthetic 1, Synthetic 2 and Synthetic 3 domains. The graphical representations were created using GeNie Software [10].

Table 2 exhibits the number of extra arcs, missing arcs and reverse arcs from the networks induced by VOGA versions in comparison to the original nets. It is perceived that VOGA using the g function as fitness function tend to produce structures closer to the originals ones. This version was the best in Synthetic 1, Synthetic 2, Synthetic 3 and Synthetic 50-200 domains. VOGA-MDL produced the same network structure produced VOGA-g when applied to Synthetic 2 and it induced the best structure when applied to Synthetic 50-100. VOGA-AIC induced a structure very similar to the one induced by VOGA-g when applied to Synthetic 3. VOGA using the two combinations of score metrics obtained structures slightly different from other versions. However, table 3 shows VOGA-"Majority vote" converged faster than the other versions.

It is possible to notice that all VOGA versions induced structures very different from the original ones when applied to Synthetic 50-100 and Synthetic 50-200. This demonstrates how hard it is to learn very dense structures having many variables. Despite of this, table 4 shows all VOGA versions obtained equivalent classification ratios for all databases demonstrating that the different score metrics have very similar influence on the classification rates.

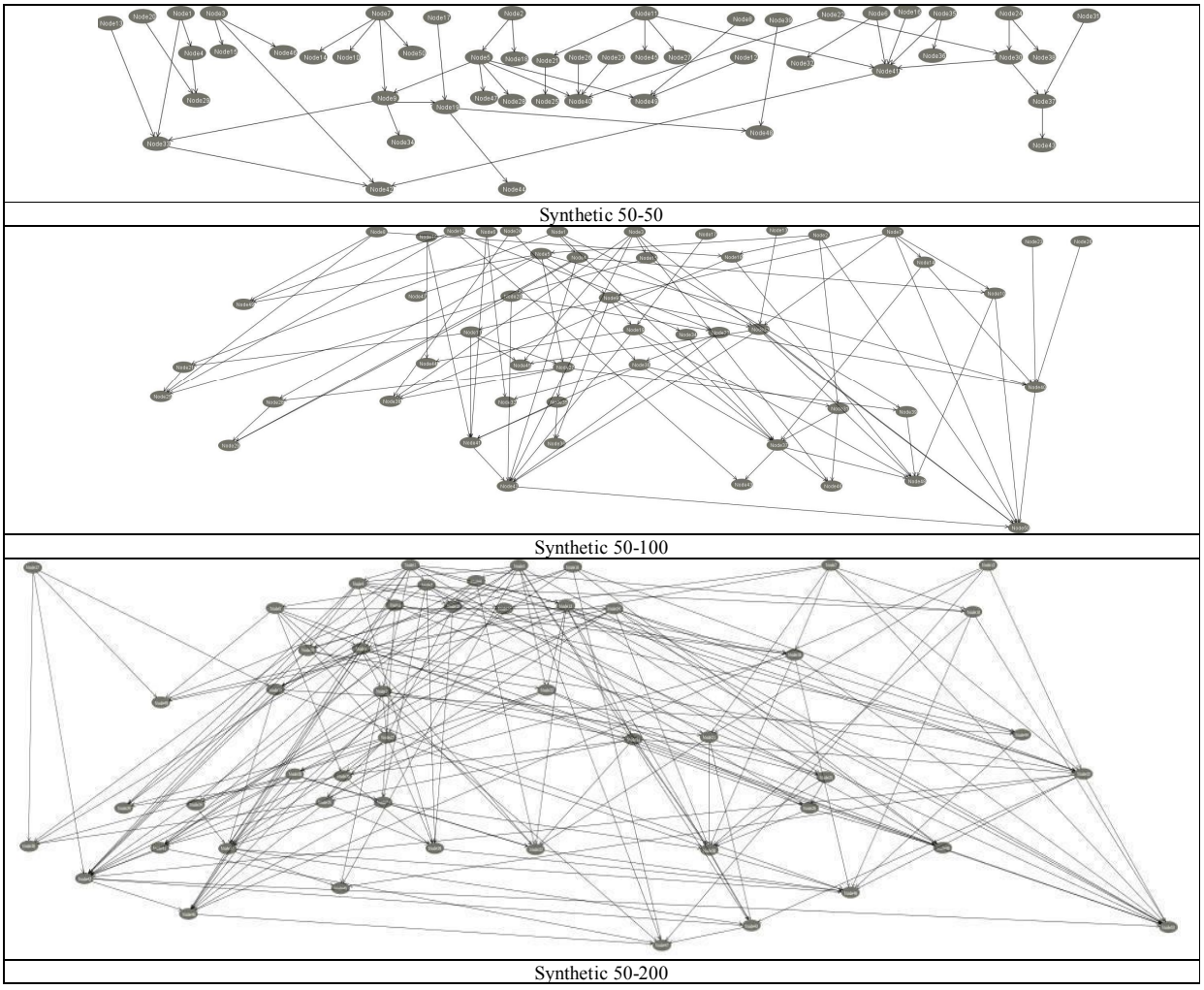

Figure 3 – Bayesian Networks representing Synthetic 50-50, Synthetic 50-100 and Synthetic 50-200 domains. The graphical representations were created using Weka Software [13].

Table 2 –. Number of extra arcs (+), number of missing arcs (-) and number of reverse arcs (R).

	VOGA-g			**VOGA-Entropy**			**VOGA-MDL**			**VOGA-AIC**			**VOGA-BDe**			**VOGA-"Sum of all scores"**			**VOGA-"Majority vote"**		
	+	-	R	+	-	R	+	-	R	+	-	R	+	-	R	+	-	R	+	-	R
Synth 1	**3**	**3**	**2**	8	7	1	3	2	4	5	1	6	6	1	7	6	3	6	7	3	7
Synth 2	**3**	**8**	**5**	8	8	6	**3**	**8**	**5**	4	8	4	**3**	**8**	**5**	**3**	**8**	**5**	**3**	**8**	**5**
Synth 3	**1**	**14**	**1**	4	14	3	0	14	4	**0**	**14**	**2**	1	15	3	1	14	2	6	14	8
Synth 50-50	18	1	18	23	2	18	25	2	19	15	1	14	**12**	**1**	**12**	18	2	14	22	2	19
Synth 50-100	66	3	24	84	5	34	**55**	**5**	**28**	84	5	34	70	2	29	85	3	32	84	5	34
Synth 50-200	**299**	**76**	**38**	305	83	34	304	78	42	306	70	41	305	84	44	304	75	41	311	81	49

Table 3 – Generation numbers necessary to convergence of VOGA algorithms using different score metrics.

	VOGA-g	VOGA-Entropy	VOGA-MDL	VOGA-AIC	VOGA-BDe	VOGA-"Sum of all scores"	VOGA-"Majority vote"
Synth 1	26	22	20	24	18	23	**15**
Synth 2	**11**	27	**11**	14	**11**	**11**	**11**
Synth 3	21	26	14	15	16	27	**11**
Synth 50-50	20	18	20	13	15	19	**12**
Synth 50-100	12	**11**	12	**11**	16	17	**11**
Synth 50-200	21	23	20	21	26	19	**13**

Table 4 – Classification Ratios obtained by VOGA.

	VOGA-g	VOGA-Entropy	VOGA-MDL	VOGA-AIC	VOGA-BDe	VOGA-"Sum of all scores"	VOGA-"Majority vote"
Synth 1	89.67	89.67	89.67	89.67	89.67	89.67	89.67
Synth 2	93.4	93.4	93.4	93.4	93.4	93.4	93.4
Synth 3	89.44	89.44	89.44	89.44	88.56	89.44	**89.52**
Synth 50-50	69.56	69.56	69.56	69.56	69.56	69.56	69.56
Synth 50-100	83.65	**83.72**	83.62	**83.72**	83.65	83.56	**83.72**
Synth 50-200	98.88	**98.98**	98.72	98.69	98.78	98.86	98.97

5 Conclusion

This paper presents five score metrics and applies them to a genetic algorithm named VOGA (Variable Ordering Genetic Algorithm) as fitness functions. The score metrics presented are: Entropy, BDe (Bayesian Dirichlet equivalent), MDL (Minimum Description Length), g function and AIC (Akaike Information Criterion). These metrics are a quality measure that can be used to estimate the probability of the Bayesian network structure being a proper mapping of a given database.

The main goal was to verify the influence caused by different metrics in the performance of VOGA. The results showed that VOGA using the g function (from K2 algorithm) tends to produce network structures closer to the original ones. However, a combination of metrics by majority vote made VOGA converge faster.

We intend to proceed along these lines of investigation by performing a study on how "Conditional Independence"-based algorithms would behave when substituting K2 algorithm in VOGA. In addition, new fitness functions may be developed in a multiobjective approach.

Acknowledgment

The authors acknowledge the support provided in conducting the experiments of this research to NACAD from COPPE/UFRJ in Brazil as well as the Brazilian research Agencies CNPq and Fapesp.

References

[1] AKAIKE, H. Information theory and an extension of the maximum likelihood principle. In B.N. Petrox and F. Caski, editors, Proceedings of the Second International Symposium on Information Theory, pages 267-281, Budapest. Akademiai Kiado. 1973

[2] BACK, T. Evolutionary algorithms in theory and practice. Oxford University Press, 1996.

[3] BOUCKAERT, R. R. Probabilistic network construction using the minimum description length principle. Utrecht Univ., Utrecht, The Netherlands, Rep. RUU-CS-94-27, 1990.

[4] BOUCKAERT, R. R. Bayesian Network Classifiers in Weka for Version 3-5-7. http://www.cs.waikato.ac.nz/~remco/weka_bn/. 2008.

[5] CAMPOS, L. M.; HUETE, J. F. Approximating causal orderings for Bayesian networks using genetic algorithms and simulated annealing. In: Proceedings of the Eighth International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU'00), v. 1, p. 333–340, 2000.

[6] CHEN, X.; ANANTHA, G.; LIN, X. Improving Bayesian Network Structure Learning with Mutual Information-based Node Ordering in the K2 Algorithm. IEEE Transactions on Knowledge and Data Engineering, vol. 20, n 5, p. 628-640, 2008.

[7] CHEN, F.; WANG, X.; RAO, Y. Learning Bayesian networks using genetic algorithm. Journal of System Engineering and Electronics, vol. 18, n. 1, p. 142-147, 2007.
[8] CHICKERING, D.M. Learning Bayesian networks is NP-Complete. In: FISHER, D.; LENZ, H (Eds). Learning from Data: artificial intelligence and statistics V. Springer-Verlag, p. 121-130, 1996.
[9] COOPER G.; HERSKOVITZ, E. A Bayesian method for the induction of probabilistic networks from data. Machine Learning, v. 9, p. 309-347, 1992.
[10] DRUZDZEL, M. J. SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: A development environment for graphical decision-theoretic models (Intelligent Systems Demonstration). Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), AAAI Press/The MIT Press, Menlo Park, CA, 1999. p. 902-903.
[11] ETXEBERRIA, R.; LARRAÑAGA, P.; PICAZA, J.M. Analysis of the behaviour of genetic algorithms when learning bayesian network structure from data. Pattern Recognition Letters, Vol. 18, No. 11-13, 11/97, 1269–1273, 0167-8655. 1997.
[12] GOLDBERG, D. E. Genetic algorithms in search, optimization, and machine learning. Massachusetts: Addison-Wesley, 1989.
[13] HALL M. et al. The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1, 2009.
[14] HECKERMAN, D; GEIGER D.; D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Mach. Learn., vol. 20, no. 3, p. 197–243, 1995.
[15] HECKERMAN, D. A tutorial on learning Bayesian networks. Local: Microsoft Research, Advanced Technology Division. Notas: Technical Report MSR-TR-95-06. 1995.
[16] HRUSCHKA Jr., E. R.; EBECKEN, N.F.F. Towards efficient variables ordering for Bayesian Network Classifiers. Data Knowl. 2007.
[17] HSU, W. H. Genetic wrappers for feature selection in decision tree induction and variable ordering in Bayesian network structure learning. Information Sciences, v. 163, p. 103-122, 2004.
[18] LARRAÑAGA, P. et al. Learning Bayesian network structure by searching for the best ordering with genetic algorithms. IEEE Trans. on Systems, Man and Cybernetics - Part A: Systems and Humans, v. 26, n. 4, p. 487-493, 1996.
[19] RUSSEL, S.; SRINIVAS, S.; AGOGINO, A. Automated construction of sparse bayesian networks. Proc. of UAI'90, p. 295-308, 1990.
[20] SANTOS, E. B.; HRUSCHKA JR., ER. VOGA: Variable ordering genetic algorithm for learning Bayesian classifiers. In: 6TH INTERNATIONAL CONFERENCE ON HYBRID INTELLIGENT SYSTEMS - HIS2006, 2006, Auckland. Proceedings of The Sixth International conference on Hybrid Intelligent Systems (HIS06). Los Alamitos CA, USA: IEEE Press, 2006.
[21] WONG, M.L.; LAM, W.; LEUNG, K.S. Using evolutionary programming and minimum description length principle for data mining of bayesian networks. IEEE Transactions PAMI, Vol. 21, No. 2, 174-178, 0162-8828. 1999.
[22] WONG, M.; LEE, S.Y.; LEUNG, K.S. A hybrid data-mining approach to discover bayesian networks using evolutionary programming, Proceedings of the Genetic and Evolutionary Computation Conference, p. 214-222, 2002.
[23] YANG, S.; CHANG, K.-C. Comparison of score metrics for Bayesian network learning. Systems, Man and Cybernetics: Part A, IEEE Transactions on 32 (3), p. 419-428, 2002.