A New Evolutionary Algorithm for Structure Learning in Bayesian Networks

A. R. Khanteymoori*, M. B. Menhaj ** and M. M. Homayounpour *
* AmirKabir University, Computer Engineering Department, Tehran, Iran.
** AmirKabir University, Electrical Engineering Department, Tehran, Iran.
Khanteymoori@aut.ac.ir, Menhaj@aut.ac.ir, Homayoun@aut.ac.ir

Abstract

A new structure learning approach for Bayesian networks (BNs) based on asexual reproduction optimization (ARO) is proposed in this paper. ARO can be essentially considered as an evolutionary based algorithm that mathematically models the budding mechanism of asexual reproduction. In ARO, a parent produces a bud through a reproduction operator; thereafter the parent and its bud compete to survive according to a performance index obtained from the underlying objective function of the optimization problem; this leads to the fitter individual. The proposed method is applied to real-world and benchmark applications, while its effectiveness is demonstrated through computer simulation. Results of simulation show that ARO outperforms GA because ARO results good structure in comparison with GA and the speed of convergence in ARO is more than GA. Finally, the ARO performance is statistically shown.

1. Introduction

In the last few years Bayesian networks (BNs) have become a popular way of modeling probabilistic relationships among a set of variables for a given domain [1,2,3]. The BN is a graphical model that denotes joint probabilistic distribution among variables of interest based on the variables’ probabilistic relationships. The structure of the BN is represented as a directed acyclic graph (DAG). In building the BN, which represents the conditional dependencies in a database of cases, the problem of searching for the structure of the BN is both important and difficult. Although sometimes experts can create good Bayesian networks from their own experience, it can be a very hard task for large domains. Therefore, many methods have been investigated to automate the creation of Bayesian networks using cases collected from past experience [4,5,6]. The automated creation of a Bayesian network can be separated into two tasks, structure learning, which consists of creating the structure of the Bayesian network from the collected data, and parameter learning, which consists of calculating the numerical parameters for a given structure. The number of possible structures is huge and it has been proved that the search is NP-hard [7]. Therefore, heuristic search procedures have been tried.

In this paper, a new structure learning method of the BNs based on asexual reproduction optimization genetic algorithm is proposed. This algorithm is a structural learning method based on free optimization algorithm inspired by one of the astounding biological phenomenon, asexual reproduction; hence we entitle it Asexual Reproduction Optimization (ARO). ARO is an individual based algorithm which intelligently guides the search process and it can reach to the global optimum in an astonishing time possessing advantages of both population and individual based algorithms. It meanwhile escapes from local optima by adaptively exploration and exploitation as inspired by the biological model. In the proposed method, BN structure is represented as chromosomes and it learns the topologies of the BN nodes.

This paper is organized as follows. In Section 2, a brief introduction to BNs is given. Section 3 introduces structure learning of the BN. ARO is completely described in Section 4. In section 5 proposed methods for BN structure learning is introduced. In Section 6, the proposed method is applied to real-world and benchmark problems and the results and analysis are shown. Finally, some conclusions are drawn in Sect 7.

2. Bayesian Networks

During the last decade, Bayesian networks (and probabilistic graphical models in general) have become very popular in artificial intelligence [1,2]. A Bayesian network (BN) provides a means of encoding the dependencies between a set of random variables, where
the random variables and dependencies are represented as the nodes and edges of a directed acyclic graph. Missing edges (which imply conditional independence) are exploited in order to factor the joint distribution of all random variables into a set of simpler probability distributions. A Bayesian network expresses a joint probability distribution over a set of random variables, and consists of: A set of random variables \( X_1, \ldots, X_n \) and a directed acyclic graph in which each variable appears once. The immediate predecessors of a variable \( X_i \) are referred to as its parents, with values \( \text{Parents}(X_i) \). The joint probability distribution is factored as:

\[
P(X_1, \ldots, X_n = x_1 \ldots x_n) = \prod_{i=1}^{n} p(X_i = x_i \mid \text{Parents}(x_i))
\]

(1)

When the variables are discrete, a tabular representation is used for conditional probabilities. For real-valued observations, Gaussian mixtures can be used. The topology of a BN gives direct information about the dependency relationships between the variables involved. In particular, it represents which variables are conditionally independent given another variable.

The process of building the BN can be separated into two tasks: structure learning and parameter learning. Structure learning is a search for an appropriate structure for the BN such that the BN accommodates the given set of samples. Parameter learning is computation of the conditional probabilities for the given BN structure such that the output of the BN approximates the distribution of the given set of samples. The most popular parameter learning method is the expectation maximization (EM) algorithm [5]. In this paper, the focus is structure learning of the BN and building an appropriate BN structure such that the BN structure accommodates the given set of samples.

### 3. Structural Learning

Consider the problem of analyzing the distribution over some set \( X \) of random variables \( X_1, \ldots, X_n \) each of which takes values in some domain \( \text{Val}(X_i) \). For simplicity, we focus on the case where the variables are discrete-valued; however our approach extends easily to the continuous case. Our input is a fully observed data set \( D = \{ x[1], \ldots, x[M] \} \), where each \( x[m] \) is a complete assignment to the variables \( X_1, \ldots, X_n \) in \( \text{Val}(X_1, \ldots, X_n) \). Our goal is to find a network structure \( G \) that is a good predictor for the data. The most common approach to this task is to define it as an optimization problem. We define a scoring function \( \text{score}(G : D) \), which evaluates different networks relative to the data \( D \). We then need to solve the combinatorial optimization problem of finding the network that achieves the highest score. For the remainder of this discussion, we take the training set \( D \) to be fixed.

Several scoring functions have been proposed; most common scoring functions are the BIC/MDL score and the BDe score [14]. The details of these scores are not relevant for our discussion. The most important property is that the scores is decomposable, i.e., that it is the sum of scores associated with individual families (where a family is a node and its parents):

\[
\text{score}(G) = \sum_{i=1}^{n} \text{score}(X_i, \text{Pa}_G(X_i))
\]

(2)

Given a scoring function, our task is finding:

\[
\arg \max_G \text{score}(G)
\]

(3)

This task is a hard combinatorial problem. Several of its specific instantiations have been shown to be NP-hard, even when the maximum number of parents per node is at most two [7]. The key intuition behind this result is that, due to the global acyclicity constraint, the choice of parent set for one node imposes constraints to the possible parent sets for the other nodes.

Robinson showed that \( r(n) \), the number of possible structures for Bayesian network having \( n \) nodes, is given by the recurrence formula as follows [8]:

\[
r(n) = \sum_{i=1}^{n} (-1)^{n-i} \binom{n}{i} 2^{i(n-i)} r(n-i) = n^{2(n-1)}
\]

(4)

Because of the super-exponential size of the search space, exhaustive search for the best structure is impossible. Many heuristic methods have been proposed for Bayesian network structure determination. We are here more specifically interested in score-based methods, primarily greedy search (GS) and minimum weight spanning tree (MWST) algorithms [9]. GS is a greedy search carried out in DAG spaces where the interest of each structure located near the current structure is assessed by means of a BIC/MDL type measurement or a Bayesian score like.

In order to learn a BN from data through heuristic search, three elements must be defined: (1) the search space, i.e. what hypotheses will be considered during learning, (2) the search heuristic, i.e. what criterion will be employed for evaluating different hypotheses, and (3) the search algorithm, i.e. how we will search for the best hypothesis. In a structure learning task, the search space is constructed from the set of all DAGs containing (as nodes) the random variables at issue, by learning the CPTs for each one of those DAGs.
4. Asexual Reproduction optimization

4.1. Biological Foundation of ARO

Asexual reproduction is a method of reproduction where a 1N (chromosome number) cell produces two to four cells with the same chromosome number. This can be done by binary fission of a motile stage or a non-motile stage. Asexual reproduction involves only one parent passing on the genetic information to their offspring. This sharing of genetic information makes the offspring identical to the parent [10]. Many species reproduce successfully in the absence of sex [11]. Asexual reproduction is common among organisms like bacteria, rapidly reproducing to generate large populations. In these large populations, mutation can provide considerable genetic variations, so sex may be less important in producing genetic diversity within the population [12].

There are different types of asexual reproduction like “binary fission”, “asexual spore production”, “plants asexual reproduction” and “budding”. Asexual reproduction can produce many plants very quickly. This is an advantage in places where the environment doesn’t change very much (bacteria). By building a large population of organisms very quickly the species is able to thrive. The great disadvantage is that when the environment changes, all of the organisms will die, if they do not have the ability to adapt to the change [13]. Eventually in the budding mathematically modeled by ARO, the parent organism produces a bud (a smaller version of itself), which eventually detaches itself from the parent and becomes a self-sufficient individual - identical to the parent. Coral also reproduces in this way, but do not detach themselves (hydra, yeast, coral, sea sponge) [13]. Asexual reproduction is a significant adaptation to specific environments and biological conditions where the cost of sexual reproduction to a species is considerable [14].

4.2. Main framework of ARO

As discussed in the previous section, there are various kinds of asexual reproduction. The proposed algorithm is inspired by the budding method of asexual reproduction. Each individual is illustrated by a binary string like the binary representation in evolutionary algorithms. A decision variable vector $X = (x_1, x_2, ..., x_N)$; $X \in \mathbb{R}^n$ is called an individual in ARO and each variable is considered such that the first bit represents the sign of the individual. The next $I_1$ bits show the integer part while the last $I_2$ bits present the decimal part of the chromosome. As a result $L = I_1 + I_2 + 1$ and the length of an individual becomes $n = L$. Figure 1 illustrates an ARO chromosome. We assume that each solution in the search space (S) is an organism in its environment. In addition, it is supposed that there are limited resources in the environment such that only the most deserving individual can survive.

![Figure 1. Chromosome with 3 parts including sign, integer and decimal parts](image)

To start the algorithm, an individual is randomly initiated in the distinctive domain of S, thereafter the individual reproduces an offspring labeled bud by a particular operator called reproduction mechanism completely described later. The parent and its offspring compete to survive according to a performance index or a fitness function. If the bud wins the competition, its parent will be discarded. Therefore, the bud is replaced with its parent and it becomes the new parent. If the parent triumphs then, the bud will be thrown away. The algorithm repeats steps illustrated in table 1 until the stopping criteria are satisfied.

It is obvious that the choice of an appropriate reproduction operator is very crucial. While ARO only applies one operator, most evolutionary algorithms use the number of operators to explore the search space and to exploit available information according to the traditional control theory. In order to reproduce, a substring in each chromosome is randomly chosen. Afterward bits of the substring mutate such that in any selected gene, 1 is replaced by 0 and vice versa. In fact, this substring named *larva* is a mutated form of its parent. According to the optimization theory, even though both exploration and exploitation mechanisms are indispensable, mutated form of its parent, only does larva explore the search space. In order to enhance the algorithm optimization ability, an exploitation mechanism is moreover appended so larva and its parent probably share their information by merging; consequently, bud is generated similar to its biological model. On the other hand, during mutation, crossover is implicitly occurred. Figure 2 shows the reproduction mechanism. After produced, the bud fitness is evaluated according to the performance index. As
illustrated in Table 1, bud fitness is compared with its parent fitness.

Table 1. Pseudo code for ARO

<table>
<thead>
<tr>
<th>Begin</th>
</tr>
</thead>
<tbody>
<tr>
<td>t=1;</td>
</tr>
<tr>
<td>% Parent Initialization between lower and upper bound</td>
</tr>
<tr>
<td>P=Initialize(L,U);</td>
</tr>
<tr>
<td>% Fitness of P is calculated</td>
</tr>
<tr>
<td>Fitness_P=fit(P);</td>
</tr>
<tr>
<td>% Stopping Criteria</td>
</tr>
<tr>
<td>While stopping conditions are not met</td>
</tr>
<tr>
<td>% P reproduces a Bud</td>
</tr>
<tr>
<td>Bud(t)=Reproduce(P);</td>
</tr>
<tr>
<td>% Fitness of Bud(t) is calculated</td>
</tr>
<tr>
<td>Fitness_Bud(t)=fit(Bud(t));</td>
</tr>
<tr>
<td>If Fitness_Bud(t) is better than Fitness_P</td>
</tr>
<tr>
<td>P=Bud(t);</td>
</tr>
<tr>
<td>Else</td>
</tr>
<tr>
<td>% Bud(t) is discarded</td>
</tr>
<tr>
<td>clear Bud(t);</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>t=t+1;</td>
</tr>
<tr>
<td>End</td>
</tr>
</tbody>
</table>

Figure 2. Reproduction mechanism generating bud chromosome

5. ARO Algorithm for BN Structure Learning

This section describes how ARO method can be applied to the structure learning of the BNs. In the ARO, the connectivity matrix is assumed to be upper triangular, and it is the same as that of the GA method. Thus, the entire solution space is searched for the fittest structure of the BN. In order to encode all of the information related to the structure learning of the BN into a chromosome and search the entire solution space for the fittest structure, an ARO representation is proposed. The BN structure with n variables is represented by an n x n connectivity matrix C = (ci, j), where

\[ C(i, j) = \begin{cases} 
1 & \text{if } i < j \\
0 & \text{otherwise} 
\end{cases} \]

and each individual of the population is encoded as a chromosome,

\[ c_1, c_2, c_3, \ldots, c_n, c_{1,2}, c_{2,3}, \ldots, c_{n-1,n}, c_{n,2}, \ldots, c_{n,n} \]

With this representation, the plain mutation operator and reproduction system would produce illegal BN structures. To overcome this problem, the connectivity matrix was assumed to be upper triangular and the connectivity matrix,

\[ \begin{pmatrix} 
0 & c_{1,2} & c_{1,3} & \ldots & c_{1,n-1} & c_{1,n} \\
0 & 0 & c_{2,3} & \ldots & c_{2,n-1} & c_{2,n} \\
0 & 0 & 0 & \ldots & 0 & c_{n-1,n} \\
0 & 0 & 0 & \ldots & 0 & 0 
\end{pmatrix} \]

was encoded as a chromosome,

\[ X = c_1, c_2, c_3, \ldots, c_n, c_{1,2}, c_{2,3}, \ldots, c_{n-2,n}, c_{n-1,n} \]

In other words, the ordering among the variables of the BN was fixed and a node Ai was allowed to have another node Aj as a parent node only if the node Aj comes before the node Ai in the ordering. This scheme restricts the values of ci, j (i ≥ j) to 0 and narrows the search space.

6. Simulation

In this section, the proposed method is applied to two realworld problems includes car diagnosis problem and ALARM Network. The validity of this method is demonstrated through computer simulation. The car diagnosis problem introduced by Norsys is a simple example of belief network. The reason why a car does not move is presumed, based on spark plugs, headlights, main fuse, etc [15]. ALARM (A Logical Alarm Reduction Mechanism) is a medical diagnostic system for patient monitoring. It is a complex belief network with eight diagnoses, sixteen findings, and thirteen intermediate variables [16].

Using these databases, the BNs were built to represent the probabilistic relationships between the variables. Theorem in [6] was utilized to apply the proposed algorithms and evaluate the validity of the structure of the given BN.

The simulation was run ten times for each problem and the parameters of the simulations were as follows:
the population size of ARO was 50 and the mutation rate was 0.05. The algorithms were stopped when 5,000 structures had been evaluated, which means that 100 generations have been evaluated.

In this simulation, we assume that there is no prior knowledge about the node ordering or the casual relationship between the nodes. Therefore, we randomly select the order of the nodes in the conventional method. Figures 3 and 4 show the learning result of the first run for the car diagnosis, and ALARM problems, respectively. And Figures 5 and 6 show the learning result of the five run for them. In these Figures the dashed lines represent the result of the conventional method and the solid lines represent the result of the proposed method.

As BN attempts to approximate the entire database, the vertical axis approaches zero, since the probability of one corresponds to zero on the negative logarithm axis. In the car diagnostic systems, it can be seen that the GA initially outperforms the ARO but after some iterations ARO outperforms the GA and continues to outperform the conventional method for the fifty generations, as shown in Figure 5. In this case ARO show good performance for less than fifty generations and it outperforms the GA method. In the ALARM network, as shown in Figure 6, it can be seen that the ARO initially outperforms the GA and continues to outperform the conventional method for the hundred generations. Statistical results are summarized in Tables 2 and 3. In these tables, the learned structures are compared with the target structure from which the sample data are taken. For further evaluation, we compare the structure results from ARO with the target network in the terms of number of arcs correctly added between the same nodes as those in the target network, the number of missed arcs of the target network, or the number of extra arcs added wrongly.
Table 2. Statistical results for Car Diagnosis Problem

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Best</th>
<th>Mean</th>
<th>variance</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>38101</td>
<td>38489</td>
<td>161136</td>
<td>37821</td>
</tr>
<tr>
<td>ARO</td>
<td>38005</td>
<td>38129</td>
<td>16810</td>
<td>37821</td>
</tr>
</tbody>
</table>

Table 3. Statistical results for ALARM Network

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Best</th>
<th>Mean</th>
<th>variance</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>74703</td>
<td>77496</td>
<td>1290145</td>
<td>73401</td>
</tr>
<tr>
<td>ARO</td>
<td>74624</td>
<td>75327</td>
<td>499296</td>
<td>73401</td>
</tr>
</tbody>
</table>

Table 4. Comparison with the target structure

<table>
<thead>
<tr>
<th>Application</th>
<th>Trial Time</th>
<th>Missed Arcs</th>
<th>Wrongly Added Arcs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car Diagnosis</td>
<td>1 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem</td>
<td>2 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 2 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 1 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alarm Network</td>
<td>1 2 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 3 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 2 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 2 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 1 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. Conclusions

In this paper, ARO, a new structure learning approach for BNs has been proposed. ARO as a new individual based optimization algorithm, inspired by asexual reproduction. Fast convergence time to the best solution, no parameter setting, being model free, evolutionary foundations and biological bases are ARO advantages making it very appropriate for real time applications. The proposed method is applied to two real-world and benchmark problems, while simulation reveals improved performance over the GA method for these case studies, according to convergence time. Indeed the ARO algorithm is faster than GA because ARO doesn’t have crossover phase.

8. References