Abstract—A new BN structure learning method using a cloud-based adaptive immune genetic algorithm (CAIGA) is proposed. Since the probabilities of crossover and mutation in CAIGA are adaptively varied depending on X-conditional cloud generator, it could improve the diversity of the structure population and avoid local optimum. This is due to the stochastic nature and stable tendency of the cloud model. Moreover, offspring structure population is simplified by using immune theory to reduce its computational complexity. The experiment results reveal that this method can be effectively used for BN structure learning.

Keywords: Bayesian network; structure learning; immune genetic algorithm

1. INTRODUCTION

Bayesian Network (BN) is an important tool for reasoning under uncertainty. The network as a whole describes the joint probability distribution of a set of random variables in the domain of interest. BNs were originally proposed by R. Howard and J. Matheson in 1981, since then they have been widely used in artificial intelligence areas such as data mining, pattern recognition, decision support, medical diagnosis, etc [1, 3].

BN learning is to identify a network structure that best reflecting the dependence relations between variables in the database of instances. BN learning includes parameter learning and structure learning in which structure learning is considerably more difficult due to its exponential complexity. Some traditional learning algorithms, such as K2 algorithm [4] and Genetic Algorithm (GA), both have its own limits. In this paper, we propose a BN structure learning method based on cloud-based adaptive immune genetic algorithm (CAIGA). The connection matrix (to be defined in Section 3) of the nodes in the network is drawn to describe a network topology and to represent an individual in the GA population. To improve the population quality and convergence efficiency, the algorithm adjusts the crossover and mutation rate adaptively based on X-conditional cloud generator according to the fitness of individual as in the traditional GA. Moreover, in order to improve the offspring’s fitness, the method introduces immune operator to vaccinate some descendants of individuals during the process of population replacement, which can also improve overall performance of the algorithm. To demonstrate the effectiveness of the algorithm, a set of experimental results are provided.
2. BAYESIAN NETWORK MODEL

Bayesian network is a Directed Acyclic Graph (DAG) which expresses the causal relationships and their strength between random variables [1]. Each node in a DAG represents a random variable and the directed arc between nodes represents the relationship between variables and is given by the conditional probability between the node and its parents. For a given BN, suppose there are \( n \) nodes such that \( X = \{X_1, X_2, \ldots, X_n\} \), then the joint distribution probability \( P \) can be expressed as follows:

\[
P(x_1, x_2, \ldots, x_n) = \prod_i P(x_i | \pi_i)
\]  

(1)

where \( x_i \) is the state value of \( X_i \) and \( \pi_i \) represents the parents of \( X_i \).

BN learning involves two processes: parameter learning and structure learning. Parameter learning is to determine the conditional probability of nodes (variables) given their parents in a network structure; Structure learning is to determine the network topology which best matches the given training data set. Commonly, if the structure is known, parameter learning can be easily accomplished by likelihood estimation when data is complete or by expectation maximization when data is incomplete. On the other hand, structure learning is much more difficult, because for a network with many variables, the possible number of network topologies could be huge. Specifically, for a network with \( n \) nodes,

\[
f(n) = \sum_{i=1}^{n} (-1)^{i-1} C_i^1 2^{n-i} f(n-i)
\]

(2)

represents the total number of possible network topology, where \( f(0)=1 \) and \( f(1)=1 \).

We can see that even with \( n=10 \), the total number of possible network structures is \( 4.175 \times 10^{18} \). It has been shown that BN structure learning is NP-hard [6], and it has been one of the central BN research in the last decades.

3. THE CLOUD-BASED ADAPTIVE IMMUNE GENETIC ALGORITHM (CAIGA)

There are several popular traditional structure learning methods such as K2 algorithm and Genetic Algorithm (GA). K2 algorithm has a fast learning speed [4], however it requires a prior knowledge on the order of nodes in the network and its performance is highly sensitive to the prior knowledge. GA is an optimization algorithm based on the biological evolution with a good parallel computing potential [11], which can be applied for structure learning. However, the traditional GA also has a defect that it adopts a fixed crossover and mutation rate, which can results in pre-maturity and slow convergence.

3.1 The principle of CAIGA

CAIGA is derived from the combination of traditional GA and the cloud model theory [7, 8]. The digital characteristic of cloud model can be expressed by \( Ex, En \), and \( He \). \( Ex \) is expectation, which describes the center of gravity. \( En \) is entropy, and represents the randomness of cloud droplets. \( He \) is the hyper entropy, showing the agglomeration of cloud droplets. CAIGA uses the average fitness of contemporary population as \( Ex \), and determine \( En \) according to “3\( En \)” principle in cloud model [9], which means that 99.74% of quantitative values contributed in the 3-sigma interval [\( Ex-3En, Ex+3En \)].
The key feature of the algorithm is that the crossover and mutation rate can be adaptively changed according to the X-conditional cloud generator.

Suppose that at the kth iteration, the fitness value of an individual \( Y_i \) is \( f_i \); the highest fitness value of the entire population is \( f_{\text{max}} \); the average fitness of the population is \( \overline{f} \); \( f^* \) is the better fitness of two cross-individuals; and \( \overline{f} \) is the fitness of mutational individual. Then the adaptive crossover rate \( p_c \) is adjusted as follows:

\[
\begin{align*}
\text{Begin} \\
Ex &= \overline{f} \hspace{2cm} \text{End}
\end{align*}
\]

\[
\begin{align*}
En &= (f_{\text{max}} - \overline{f}) / c_1 & \text{// } c_1 \text{ is a control parameter} \\
He &= En / c_2 & \text{// } c_2 \text{ is a control parameter} \\
En^* &= \text{normrnd}(En, He) \\
p_c &= 0.9 - 0.5 \exp(-(f^* - Ex)^2 / (2(En^*)^2)) & \text{//X-conditional cloud generator}
\end{align*}
\]

Similarly, the process of the adaptively adjusted mutation rate \( p_m \) is done as follows:

\[
\begin{align*}
\text{Begin} \\
Ex &= \overline{f} \hspace{2cm} \text{End}
\end{align*}
\]

\[
\begin{align*}
En &= (f_{\text{max}} - \overline{f}) / c_3 & \text{// } c_3 \text{ is a control parameter} \\
He &= En / c_4 & \text{// } c_4 \text{ is a control parameter} \\
En^* &= \text{normrnd}(En, He) \\
p_m &= 0.5 \exp(-(f - Ex)^2 / (2(En^*)^2)) & \text{//X-conditional cloud generator}
\end{align*}
\]

where \( En \) has an impact on the cliffy degree of the normal cloud; \( He \) determines the discrete degree of the cloud drop.

### 3.2 The Integration of Immune Theory and GA

The immune theory has been introduced into GA to inhibit the degeneration of population in the process of optimization and to improve the overall performance of the algorithm [10]. The idea is to first extract some immune genes to develop a vaccine based on the prior knowledge, and then choose some individuals in the population to vaccinate at a certain probability. At each iteration, the individuals which have been vaccinated are examined. If an individual’s fitness value is smaller than its parent’s, this implies that degeneration might have happened during crossover and mutation. In that case, the new individual is replaced by its parent. Otherwise, the new individual will be part of the next generation of population.
4. BN STRUCTURE LEARNING USING CAIGA

4.1 Coding Design and Operation of Crossover and Mutation

We adopt the connection matrix of nodes $E = (C_{ij})$ to represent the network structure [11]. $C_{ij}=1$ means node $i$ points to node $j$; $C_{ij}=0$ means there is no link between node $i$ and node $j$. A simple BN shown in Fig. 1 has a connection matrix given below:

$$E = \begin{bmatrix}
0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}$$

In crossover operation, we select a cross bit of two individuals at random every time, and exchange the two cross bits according to adaptively adjusted crossover rate $p_c$. In mutation operation, we randomly choose a position $(i,j)$ according to the varied mutation rate $p_m$, and exchange $C_{ij}$ from 0 to 1, or from 1 to 0 (that’s adding an edge, removing an edge, or reversing an edge).

4.2 Initial Population

It’s well known that the “quality” of initial population has a profound impact on the convergence rate. We first calculate the weight between nodes, and then generate a candidate network using the Maximum Weight Spanning Tree (MWST) [12]. We then randomly add an edge, remove an edge, or reverse the direction of an edge to the candidate network to produce a set of suitable structures as the initial population.

4.3 Fitness Function

Fitness function is used to measure the closeness between a Bayesian network structure $B$ and the data set $D$. We choose the BIC (Bayesian Information Criterion) function, which is one of the scoring functions used in BN structure learning algorithms. The BIC function is expressed as follows:

$$Q_{\text{BIC}} = LL(B|D) - \frac{1}{2} \log N \times \text{Dim}(B)$$

where $LL(B|D) = \sum_{i=1}^{n} \log P(B|D)$ is the measure of the number of bits which data $D$ requires; $1/2\log N$ represents the number of bits used for each parameter; and $\text{Dim}(B) = \sum_{i=1}^{n} (r_i - 1)q_i$ is the dimension of BN, i.e., the number of free parameters needed.
4.4 Detect and Modify Network Structure

At the end of each iteration, newly generated individuals may include some illegal structures. For example, a bi-directional arcs or cycles could be present in the network topology. Both of them violate the assumption of a DAG. Therefore illegal individuals such as that should be detected and modified. When a bi-directional arc is detected, i.e., there is a symmetric connection in the connection matrix, \( C_{ij} = C_{ji} = 1 \), we will randomly set \( C_{ij} = 0 \) or \( C_{ji} = 0 \). If there is a cycle in the structure, we will randomly select one of the arcs in the loop to remove or reverse.

4.5 The CAIGA Algorithm

To find the optimal solution more quickly, we record the individual having the highest fitness value. Suppose \( l_{inv} \) is the consecutive invalid-number of iteration, when the maximum fitness value remains the same after \( l_{inv} \) consecutive generations, or the total number of iterations exceeds the maximum number of iterations, the process stops. The selection of vaccines is based on the domain knowledge provided by the subject matter experts. Fig.2 shows the pseudo-code of CAIGA.

**Step 1** Initialize population size \( N_{pop} \), iteration number \( loop \), consecutive invalid-number of iteration \( l_{inv} \), and maximum iteration number \( l_{max} \).

**Step 2** Generate initial population using MWST.

**Step 3** Extract the vaccine.

\[
\text{for} \ (i = 1; \ i < N_{pop}; \ i++) \ \{ \\
\quad \text{Vaccinate individual } p_i; \\
\}
\]

**Step 4** Compute fitness value of individual.

**Step 5** if \( (loop < l_{max} \ || \ l_{inv} > 20) \)

Go to **Step 6**;

else Go to **Step 9**;

**Step 6** Iterate using CACG.

Detect and Modify illegal DAGs.

**Step 7** Offspring selection and population replacement.

**Step 8** \( loop = loop + 1; \)

Go to **Step 5**;

**Step 9** Find the individual having the greatest fitness.

It represents the final solution.

---

Figure 2. The pseudo-code of CAIGA
4.6 Numerical Experiments

In this section, we choose a simple network ASIA and a Car Diagnosis network to demonstrate the effectiveness of CAIGA in BN structure learning. AISA is a popular bench-mark network with 8 binary nodes and 8 edges as shown in Fig.3. The Car Diagnosis network, given in Fig.4, is another artificial network with 20 nodes and 22 edges [17].

![Figure 3. The ASIA network](image)

![Figure 4. Car Diagnosis network structure](image)

In this experiment, we generated a dataset of 1,000 cases for the ASIA network based on standard sampling algorithm and a dataset of 10,000 cases for Car Diagnosis network using Netica tool. We choose $c_1 = c_3 = 3.0$, $c_2 = c_4 = 10$ to produce adaptively adjusted crossover and mutation rate in CAIGA.

In ASIA structure learning, we choose the population size to be 10; consecutive invalid-number of iteration is 20; and maximum iteration number is 100. We use different subsets of the data (first 200, 500, and the entire 1,000 cases) in 10 Monte Carlo trials. The optimal ASIA structure CAIGA obtained is shown in Fig.5. The average number of iteration, extra edges, missing edges, and reversed edges are recorded in Table I.

![Figure 5. The optimal learned structure of ASIA](image)
TABLE I. Comparison of ASIA Structure Learning under Different Data Sets

<table>
<thead>
<tr>
<th>Sets of Data</th>
<th>Iteration Number</th>
<th>Added Edges</th>
<th>Removed Edges</th>
<th>Reversed Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>22.8</td>
<td>0.2</td>
<td>2.0</td>
<td>0.1</td>
</tr>
<tr>
<td>500</td>
<td>18.3</td>
<td>0.1</td>
<td>1.5</td>
<td>0.1</td>
</tr>
<tr>
<td>1000</td>
<td>13.1</td>
<td>0.1</td>
<td>1.0</td>
<td>0</td>
</tr>
</tbody>
</table>

As we can see from Table I and Fig.5, CAIGA performs reasonably well. Furthermore, as the data samples increases, the number of extra edges, missing edges, and reversed edges decreases, and the convergence time of the algorithm to obtain the optimal solution is also reduced.

In order to compare with the traditional GA and the I-GA proposed in [10], we apply GA, I-GA, and CAIGA for learning the ASIA network. With 1,000 ASIA data sets, we select the population size \( N_{\text{pop}} \) to be 10 and 20 in two different trials, and let the iteration number be 50. Fig.6 and Fig.7 show the relationships between iteration time and the highest fitness of the population for each algorithm. It can be seen that CAIGA performs noticeably better than I-GA and the traditional GA.

Figure 6. Comparison in ASIA structure learning (\( N_{\text{pop}} = 10 \))

Figure 7. Comparison in ASIA structure learning (\( N_{\text{pop}} = 20 \))
For the Car Diagnosis structure learning, we choose population size $N_{\text{pop}}$ to be 30; consecutive invalid-number of iteration be 50; and the maximum iteration number be 200. We apply different subsets of the data (first 3,000, 5,000, and the entire 10,000 cases) for structure learning, each with 10 Monte Carlo trials. The optimal learned structure for Car Diagnosis network is shown in Fig.8. The average number of iteration, extra edges, missing edges, and reversed edges are recorded in Table II. Again, similar performance behavior as in the ASIA network case is observed.

![Figure 8. The optimal learned structure of Car Diagnosis](image)

<table>
<thead>
<tr>
<th>Sets of Data</th>
<th>Iteration Number</th>
<th>Added Edges</th>
<th>Removed Edges</th>
<th>Reversed Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>113</td>
<td>3.6</td>
<td>6.7</td>
<td>1.7</td>
</tr>
<tr>
<td>5000</td>
<td>109</td>
<td>2.2</td>
<td>5.9</td>
<td>1.3</td>
</tr>
<tr>
<td>10000</td>
<td>82</td>
<td>1.6</td>
<td>3.8</td>
<td>0.4</td>
</tr>
</tbody>
</table>

In the last experiment, we separately use I-GA and CAIGA to learn the Car Diagnosis network structure with 10,000 cases. The population size $N_{\text{pop}}$ is 30. Fig.9 shows the results of the first 150 iterations for each algorithm.

![Figure 9. Comparison in Car Diagnosis structure learning](image)
From the results of the above experiments, it is clear that the newly proposed algorithm is effectively in BN structure learning. With the same sets of data samples, GA appears premature and converges to local optimum, while CAIGA not only finds the global optimum, but converges faster than that of I-GA [10]. Overall, we found that CAIGA has a good potential in BN structure learning with both better performance and faster convergence rate.

5. CONCLUSION

A new method called CAIGA is proposed for BN structure learning. The method is based on the concept of cloud theory and immune Genetic Algorithm. We have validated the method with extensive experimental results. The experiment results indicate that this new algorithm has a better convergence speed and has a potential to avoid local optimum. However, in general, BN of different topologies may have identical BIC value. We call these structures Markov-equivalent. It would be an interesting future research direction to examine how to identify the best structure in a set of equivalent models.

REFERENCES