

A novel method for Bayesian networks structure learning based on Breeding Swarm algorithm

Ali Reza Khanteymoori¹ · Mohammad-H. Olyaei¹ · Omid Abbaszadeh¹ · Maryam Valian²

Published online: 6 April 2017
© Springer-Verlag Berlin Heidelberg 2017

Abstract Bayesian networks (BNs) are widely used as one of the most effective models in bioinformatics, artificial intelligence, text analysis, medical diagnosis, etc. Learning the structure of BNs from data can be viewed as an optimization problem and is proved that this problem is NP-hard. Therefore, heuristic methods can be used as powerful tools to find high-quality networks. In this paper, an interesting approach which is based on Breeding Swarm has been used to learn BNs. Breeding Swarm is a hybrid GA/PSO which enable us to benefits the strengths of particle swarm optimization with genetic algorithms. In order to assess the proposed method, several real-world and benchmark applications are used. Results show that our method is a clear improvement on genetic algorithm and particle swarm optimization.

Keywords Graphical models · Bayesian networks · Structure learning · Bio-inspired algorithm · Particle Swarm optimization · Breeding Swarm

Communicated by V. Loia.

✉ Ali Reza Khanteymoori
khanteymoori@znu.ac.ir

Mohammad-H. Olyaei
mh.olyaei@znu.ac.ir

Omid Abbaszadeh
o.abbaszadeh@znu.ac.ir

Maryam Valian
M.valian@kiaua.ac.ir

¹ Department of Computer Engineering, University of Zanjan, Zanjan, Iran

² Islamic Azad University, Karaj Branch, Karaj, Iran

1 Introduction

Nowadays, Bayesian networks (BNs) become a considerable probabilistic models in the field of artificial intelligence and widely applied in many areas such as computational biology, medical diagnosis, vision recognition, data mining, information retrieval and so on (Ahmad et al. 2012; Hill 2012; Ji 2011; Li 2011; Murphy and Mian 1999; Yang et al. 2010; Zou and Conzen 2005). The BN is a graphical model which is represented as a directed acyclic graph (DAG) where nodes correspond to variables and arcs denote dependences among variables which encodes the joint probability distribution. The capability of representing the uncertainty of real problems, comprehensibility of graphical model, the strong mathematical bases of formalism, inference the network and calculating the value of unobserved variables based on observed variables are some of the main advantages of using BNs (Gámez et al. 2011; Ji 2011).

As the popularity of BNs increased, the building of BNs has been considered by researchers. The creation of BNs can be divided into two categories: Structure learning and parameter learning. The former refers to earning the topology of network based on the collected data, and the later refers to calculating the conditional probabilities of a given structure.

The structure learning can be done as a manual task by an expert but this is so complicated and time-consuming and moreover sometimes it might be inaccessible. On the other hand, it has been proved that structure learning is an NP-hard problem (Chickering 1996). Therefore, the structure learning based on the available data is increasingly becoming a vital factor in using BNs.

Generally, there are two main approaches for structural learning: dependency-based and score-based approaches (Gámez et al. 2011). The first contains methods which estimate dependencies among variables and the second refers

to approaches which try to find a network structure that is most fit with training data. The score-based method can be seen as optimization problem because there is an evaluation function which gives score to candid structures. The goal is to finding the structure which maximizes the function measure (Khanteymoori et al. 2011). Although dependency-based approaches are relatively simple for implementations, the assessment of dependencies is complicated and unreliable when the number of variables is increased (Ji et al. 2013). Therefore, score-based methods are attracting considerable interest due to learning structure.

Since this problem is NP-hard, Several authors have attempted to solve the problem based on heuristic methods which can be refer to ant colony optimization (Ji 2011), asexual reproduction optimization (ARO) (Khanteymoori et al. 2011), particle swarm optimization (Cowie et al. 2007), artificial bee colony algorithm (Ji et al. 2013), genetic algorithm (You 2001; Larrañaga 1996; Wong et al. 1999) and so on (Alonso-Barba and Puerta 2011; Gámez et al. 2011; Larrañaga 2013; Ziegler 2008).

The aim of this study is to provide a suitable method for structure learning by using a hybrid GA/PSO algorithm called Breeding Swarm Particle Swarm Optimization (BSPSO). In this method, a portion of population which is called breeding rate is reproduced according to crossover operator in genetic algorithm (Matteu and Terence 2006). The combination of these algorithms allows to increase the chance of reproducing particles with higher scores, while the cost of computations is not changed significantly. Furthermore, by using mutation operator, the diversity of particles will be saved and will prevent from early convergence. Therefore, the search strategy will be more robustness against local maxima.

This paper is organized as follows: In Sects. 2 and 3, some preliminaries and basics concepts about BNs and structure learning are reviewed. In Sect. 4, our method is described in detail. Results are shown in Sect. 5, and our conclusions are drawn in the final section.

2 Bayesian network

BNs or causal networks represent probability distribution, correlations and relations between random variables in real-world problems. We can suppose BN as a tuple $BN(S, \theta)$ in which S is structure which demonstrates the random variables and the relations between them by a directed acyclic graph (DAG). In a DAG, random variables and relations are nodes and edges, respectively. θ is a set of local parameters which denotes the conditional probability distributions for the values of each variable according to the structure S . These measures are saved as a table for each variable which is called conditional probability table (CPT).

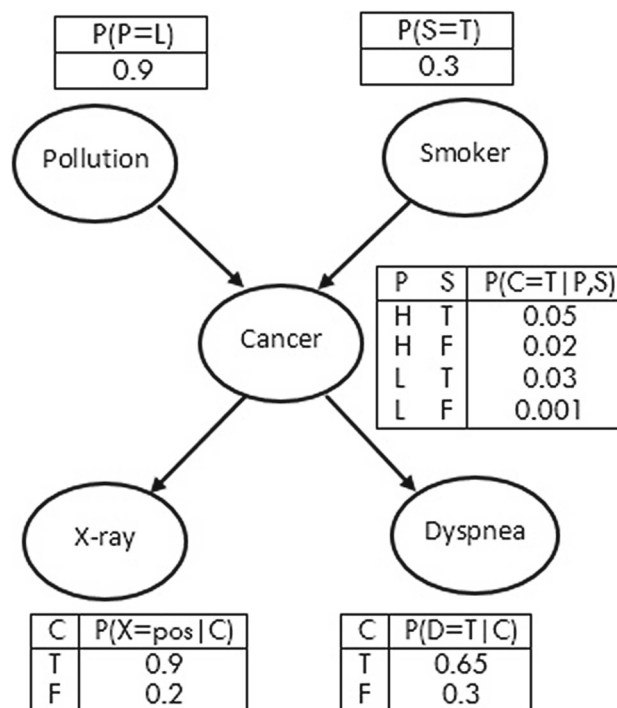


Fig. 1 An example of a simple Bayesian network structure

Figure 1 shows a simple BN with five variables and their CPTs which all variables are binary. The set of variables contains Cancer, Smoker, Pollution, X-ray and Dyspnea.

The information about the variables and the dependencies between them can be gained easily from the topology of a BN.

Suppose a BN with a set of random variables X_1, \dots, X_n . The immediate predecessors of variable x_i are referred to as its parents, with values $parents(x_i)$. The joint probability distribution is factored as:

$$P(X_1 = x_1, \dots, X_n = x_n) = \prod_{i=1}^n p(X_i = x_i | parents(x_i)) \tag{1}$$

As mentioned earlier, in order to design a BN, two phases must be accomplished. First, search for suitable structure and next, compute CPTs of variables based on the found structure such that the output approximates distribution of the given set of samples. The both of these phases are important and are called structure learning and parameter learning, respectively.

The most popular parameter learning method is the expectation maximization (EM) algorithm (Chickering et al. 1995). This paper concentrates on the structure learning and presents a new approach to find an appropriate BN structure which is matched with the given dataset.

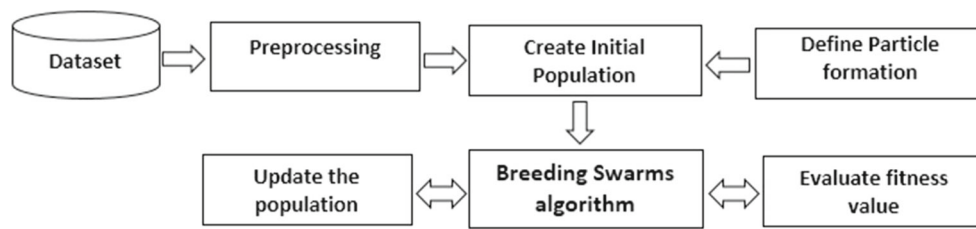


Fig. 2 The diagram of the proposed method

3 Structure learning

The structure learning for a BN can be stated as follows: X is a set of random variables X_1, \dots, X_n . Each variable X_i has a special domain which takes values $val(X_i)$. Given a training set with M cases, $D = \{d[1], \dots, d[M]\}$ where $d[i]$ is an instance of domain variables $val(X_1, \dots, X_n)$. The learning goal is to find a network structure G that is a good predictor for the data.

It has now been demonstrated that the number of possible structures for BN with n nodes can be given by the recurrence relation as follows (Robinson 1977):

$$r(n) = \sum_{i=1}^n (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} r(n-i) \in n^{2^{o(n)}} \quad (2)$$

The above formula obviously shows that the size of search space is exponential and it is impossible to use an exhaustive search to find the best structure. As already mentioned, dependency based and score based are two main structure learning methods. In this article, we are only interested in the score-based approach. Also, the reasons that why score-based approach are more popular than dependency-based approach have been declared.

Score-based approaches define structure learning as an optimization problem. In order to find the best structure that fits with training data, use a scoring criteria and a search algorithm. several scoring criteria have been proposed so far that can be denoted to Bayesian information criteria (BIC), minimum description length (MDL) score and Bayesian Dirichlet (BDe) score (Heckerman et al. 1995). Since the search space is exponential and the search problem is NP-hard, using the heuristic methods is popular.

Among the heuristic methods, genetic algorithm (GA) has been widely noticed. However, one of the major drawbacks to exploiting this algorithm is high computational complexity especially when the number of random variables are increased. Using PSO can resolve the challenge of computational complexity, but this method suffers from premature convergence when diversity of the population is low. In the proposed method, in order to exploit the advantages of GA and PSO altogether, a hybrid algorithm called Breeding Swarms PSO (BSPSO) algorithm is used.

4 Proposed method

The input is a dataset which is a $m \times n$ matrix M . m is the number of records, and n is the number of variables. The output is a BN with n nodes that is completely fittest with data. In the first step, some information in matrix M is modified with preprocessing. Next, through the use of BS, we were able to find the best BN structure. Figure 2 shows the diagram of the proposed method. Moreover, some important steps will be introduced in details as follows.

4.1 Preprocessing

In the initial stage of the preprocessing, continuous variables will be discretized. For example, suppose age as a continuous variable. It can be discretized to less than 20, between 20 and 45 and more than 45. Next, if there is a field with missing values, the measure ‘ U ’ will be added to it and its variable too. Finally, all the variables have discrete domains without missing values.

4.2 Breeding Swarms algorithm

Breeding Swarms or GA/PSO is a hybrid algorithm which has been proposed by Settles (Matthew and Terence 2006). This algorithm combines the updating rules of PSO including particle velocity and particle position with the operators of GA. The reproduction between the particles increases the probability of making more desirable particles in search space. Since only the portion of population is constructed by GA, the computational complexity is improved. Moreover, the diversity of population prevents from the early convergence. ψ is the only additional parameter that is called breeding rate and determines the portion of combination. Its measure depends on the problem criteria and is set experimentally. The algorithm is summarized in Fig. 3. Suppose N is size of the population. The position of $(N - 1)\psi$ particles is updated based on PSO algorithm and construct $(N - 1)\psi$ particles of next generation. The remains are made by GA from all the particles of current population.

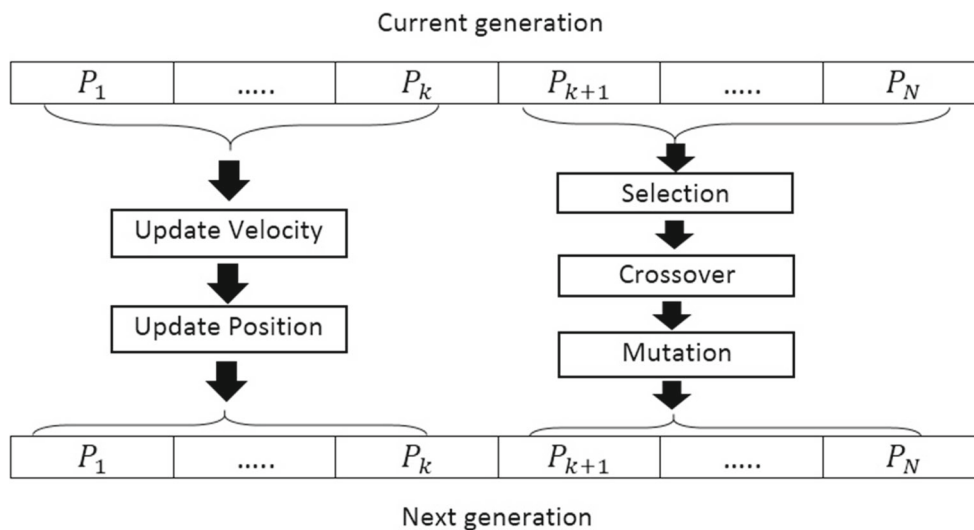


Fig. 3 The Breeding Swarm PSO algorithm

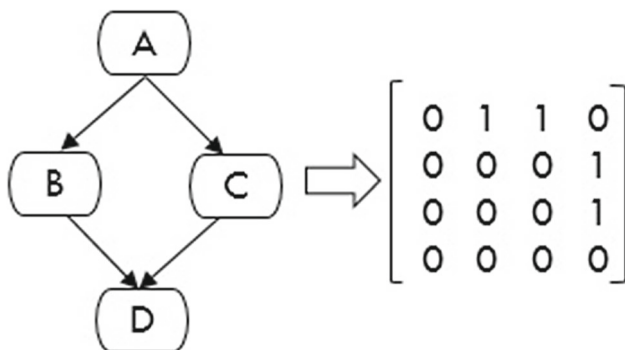


Fig. 4 The corresponding particle of a simple BN

4.2.1 Particle formation

Suppose there are N random variables. Since each BN structure is a directed acyclic graph, it can be represented by an adjacency matrix, therefore, each particle can be considered as a $N \times N$ matrix. Figure 4 is an example and displays the corresponding particle of a BN.

4.2.2 Initial population

A population is composed of N particles and can be randomly generated. Each cyclic graph with n nodes can be a candidate structure for BN.

4.2.3 Selection operator

Binary Tournament selection is used to generate $N\psi$ particles of the next generation. For selection of each parent, two particles are randomly selected from the whole of population. Then, the particle with better score is selected. By having this

strategy, weak particles have a considerable chance for survival.

4.2.4 Recombination operator (crossover)

In the proposed method each chromosome is coded in matrix form. Therefore, single point crossover is adopted with probability P_c to determine whether the operation will be performed or not. The operator randomly selects a crossover point, and the bit strings after that point are swapped between the two parents.

This recombination may lead to unjustified offspring and make cycles in the corresponding graphs. Therefore, checking the justifiability of the two produced offspring and correcting them is indispensable.

As it is illustrated in Fig. 5, the cycles in graphs can be classified in to three groups. Amending the unjustified offspring is done according to the type of cycles:

- Cycle with length one: when there is a cycle from one node to itself. In this case, some entries of adjacency matrix in the main diagonal are one. Since both of the parents are justified and the main diagonal of both of them is zero, after applying the crossover operator, there is not any cycle with length one in new offspring.
- Cycle with length two: as it is shown in Fig. 5, there is a bi-directed edge between two nodes. In this case, there are two entries with measure one which are symmetric with respect to the main diagonal. In this case, in order to improve the efficiency of the proposed method, the score of each edge is computed individually based on the Bayesian criteria and the edge that has minimum score against others will be deleted.

Fig. 5 Type of graph cycles. (1) length one, (2) length two and (3) length three and more

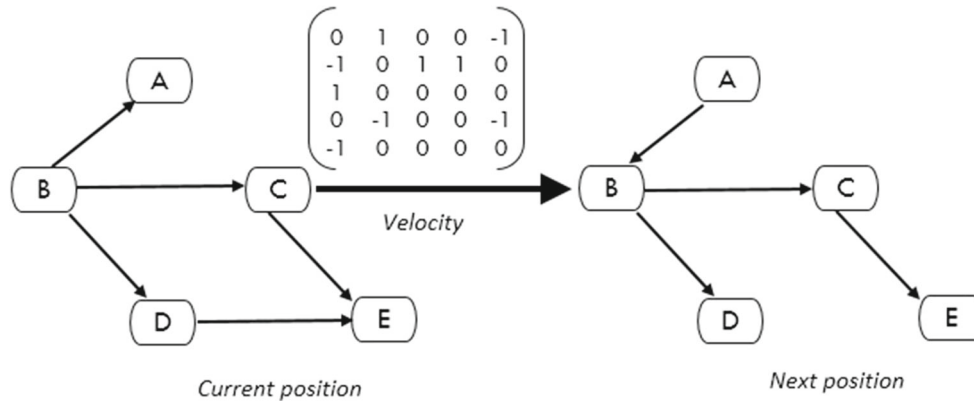
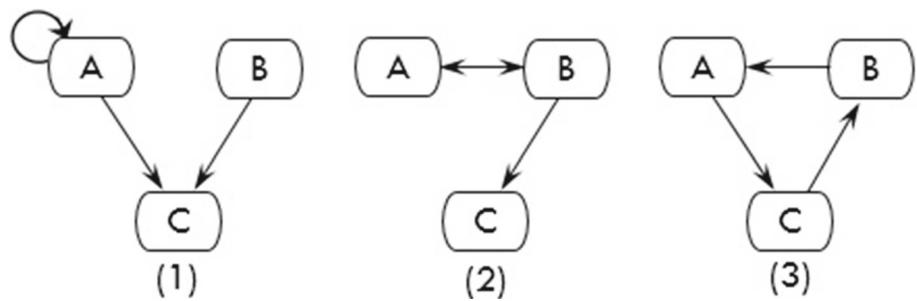


Fig. 6 An example of computing the new position of a particle

- Cycle with length three or more: the Warshalls algorithm with the order $O(n^3)$ is used to find cycles with length more than two. Although this algorithm has been proposed to find the shortest path between two nodes, it can be used to find the shortest path between one node and itself. In the proposed method, among the edges which construct the cycle, the edges with least score will be deleted.

4.2.5 Mutation operator

This operator randomly inserts or deletes an edge into the structure (graph) or selects an edge and reverses its direction. The insertion of a new edge may lead to creating a cycle in the mutated chromosome. In this case, the inserted edge will be deleted.

4.2.6 Update the position of particle

This operator computes the new position for each particle by applying its velocity (V) to its current position. Each point in search space is a Directed Acyclic Graph (DAG) and is represented in matrix form. Hence, it is appropriate that its velocity is denoted in the form of a matrix $n \times n$ which n is the number of random variables. Let V be a set of velocities $V = \{v_1, v_2, \dots, v_m\}$ that each $v_{ij}^m \in \{0, 1, -1\}$. m is the size of population. If $v_{ij} = 1$, it means an edge is added from i to j . If $v_{ij} = -1$, it means the edge between i and j is

deleted, and $v_{ij} = 0$ means that the edge between i and j is not changed. v_i is computed based on the current velocity, position of the best particle and best position that the i th particle has seen up to now.

The insertion of an edge may lead to a cycle in the corresponding graph of particles. In this condition, in order to decrease the computational complexity, the inserted edge will be removed. The Fig. 6 illustrates an example of applying velocity to the current position of a particle.

The proposed method is summarized in Fig. 7.

5 Experimental results

In order to evaluate the proposed method, several real-world problems are used and the validity of this method is demonstrated through computer simulation.

(A) Datasets

In this experiment, seven benchmarks are used which are described as follows:

- (1) ASIA network: This network has been represented by Lauritzen and Spiegelhalter (Lauritzen and Spiegelhalter 1988) which is used as a basic model for analyzing the performance of structure learning algorithms. This network is composed of 8 nodes and 8 edges which is considered as a small network.
- (2) Car diagnosis problems: This problem has been introduced by Norsys (NorsysSoftwareCorp) and is a


```

Algorithm BS_BN
Set N=Swarm size,ψ=Breeding ratio;
for each particle do
    Initialize position  $G_{i,0}$ 
end{for}
Repeat
    Sort(Swarm,Score);
    for i=1to N(1-ψ) do
        Update Velocity of particle  $P_i$  Using  $G_t^g, G_{i,t}^p$  ;
        Update  $G_i$  Positions of particle  $P_i$  ;
    end{for}
    for i=1+N(1-ψ) to N do
        Select Parents from whole swarm;
        Offsprings=Crossover(Parents);

         $G_i$ =Mutation(offsprings,Mutation Ratio) ;
    end{for}
    Update  $G_{t+1}^g, G_{i,t+1}^p$  ;
until stop criterion is satisfied
output  $G^g$ ;
End.
    
```

Fig. 7 The pseudo-code of the BPSO method

simple example of a belief network. The reason why a car does not move is presumed based on spark plugs, headlights, main fuse, and so on. 15 nodes as well as 17 edges are used in this BN. Moreover, all nodes of the network take discrete values. Some of them can take on three discrete states, and the others can take on two states. A database of two thousand cases is utilized to train the BN. The database was generated from Netica tool.

- (3) Child network
This network represents a diagnosis model of heart diseases for newborn babies which has composed of 20 nodes and 25 edges (Spiegelhalter and Cowell 1992).
- (4) Mildew network
This network is designed for determining the amount of fungicides against mildew of wheat (Jensen and Jensen 1996) and consists of 35 nodes and 46 edges.
- (5) Insurance network
Insurance is a medium network with 27 nodes and

52 edges which has been proposed for estimating the risks of car insurance (Binder 1997).

- (6) ALARM network: A logical alarm reduction mechanism (ALARM) is a medical diagnostic system for patient monitoring. It is a complex belief network with 37 nodes and 46 edges (Beinlich 1989). As in the previous example, the database is generated from Netica tool.
- (7) Hailfinder network
The network has been represented for forecasting stormy weather in Northeastern Colorado (Abramson 1996). Hailfinder composed of 56 nodes and 66 edges.
It is noted that all of the introduced benchmark networks are available in <http://www.bnlearn.com/bnrepository/>. The goal of the structure learning for BN is to obtain the structures which are close to the desired networks.
- (B) Evaluation criteria
Quantitative and Qualitative criteria are two main standards which are used to analyze the performance of the output results. The former refers to Bayesian Dirchlet metric which has been used in K2 algorithm (Cheng et al. 1997; Cowell 1998) and its score is between 0 and $-\infty$. The later compares the output with desired structure and is based on the differences evaluates the output structure. Structure learning factor (SLF) and topology learning factor (TLF) (Colace et al. 2010) are two popular qualitative criteria which are defined as follows:

$$\begin{cases} \text{SLF} = \frac{\text{TC}}{\text{TE}} \\ \text{TLF} = \frac{\text{TC} + \text{IE}}{\text{TE}} \end{cases} \quad (3)$$

In the above formula, TC is the number of edges which are correctly determined, TE is the number of edges in target graph and IE is the number of edges which determined correctly but with wrong direction.

- (C) Simulation results
In order to set used parameters optimally in proposed method, several experiments are done based on the seven available databases described in Table 1.
- Breeding rate:
In this experiment, the goal is find the optimal breeding rate for each problem. For this purpose, all other parameters are fixed. In order to evaluate performance of the algorithm, the Bayesian Dirichlet metric which mentioned previously is used. The measures of bellow table are according to the average of 10 times running for each problem.
As it is shown in Fig. 8, the quality of scores when

Table 1 The default parameters in experiments

Standard Bayesian network	Size of network	Size of population	Number of records
Asia	8	20	5000
Car	15	20	10,000
Child	20	15	1000
Insurance	27	15	1000
Mildew	35	10	1000
Alarm	37	10	3000
Hailfinder	56	10	10,000

the breeding rate is located in range [0.1, 0.5] is better than second part, i.e. [0.5, 0.9]. Therefore, it can be said that when GA is dominated to PSO, the performance of method will be decreased. On the other hand, when the breeding rate is less than 0.3, the performance has been decreased that shows that the best range for breeding rate is [0.3, 0.5]. The experimental results in Table 2 demonstrate that the algorithm has different behavior against other databases and the performance is more suitable when the breeding rate is greater than 0.5. The experiment reveals that the number of variables in databases has direct impact such that when the BN is small, the diversity of the population respect to its size is suitable but when the size of BN is increased, it is necessary to have more diversity in population and search space is explored more accurately. The experimental results show that the range [0.6, 0.8] is suitable in Alarm and Car problems.

• Size of the population

The other parameter which should optimally set is the size of population. For this purpose, the breeding rate is set based on the results of pervious experiment. In this test, the size of population is changed from 5 to 40 and in each phase, 5 particles are inserted to the population. The algorithm is run 10 times and the average of gained scores is computed.

Table 2 Experimental results of breeding rate setting in three test Bayesian networks

Breeding rate (Ψ)	The average of structure scores		
	Asia	Car	Alarm
0.1	-2376.40	-5336.3	-10,389.2
0.2	-2376.29	-5334.3	-10,321.4
0.3	-2376.02	-5328.8	-10,258.4
0.4	-2375.86	-5326.6	-10,273.0
0.5	-2375.98	-5316.6	-10,182.2
0.6	-2376.40	-5312.5	-10,131.9
0.7	-2376.58	-5312.5	-10,103.7
0.8	-2376.68	-5316.7	-10,118.2
0.9	-2377.08	-5315.4	-10,111.9

The results are listed in Table 3. As it is shown, in Car and Asia problem, when the size of population is 20, the performance is suitable. But in Alarm network, with increasing the size of population, the resulted scores are improved. However, the growth of population leads to the more computational complexities. Therefore, it is appropriate that the size is determined based on the available resources.

• Performance evaluation

For further assessment, the proposed method is run 30 times with the optimal parameters.

The results are depicted in Figs. 9, 10 and 11 for Asia, Car and Alarm networks, respectively.

The formulas 4–7 are used to compare the results with the target networks.

$$\frac{\text{Total number of TC edges}}{\text{Total number of edges}} \tag{4}$$

$$\frac{\text{Total number of IE edges}}{\text{Total number of edges}} \tag{5}$$

Fig. 8 The diagram of breeding rate for ASIA Bayesian network

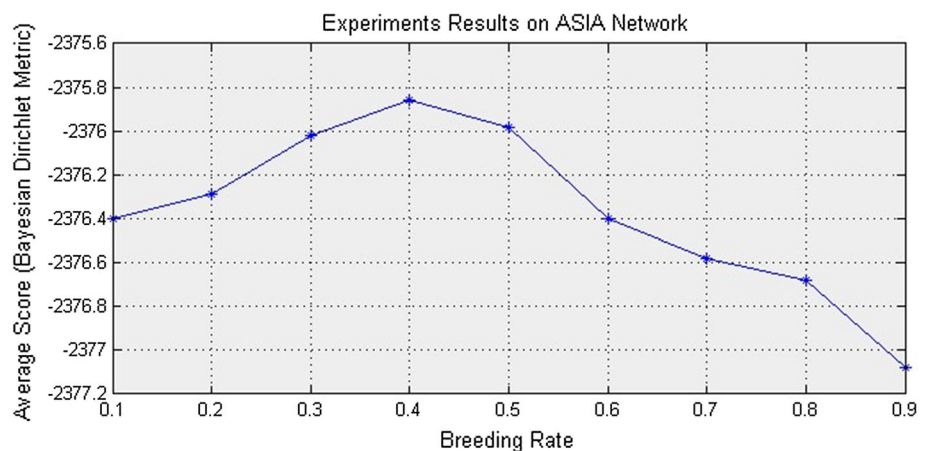


Table 3 Experimental results of population size setting in three test Bayesian networks

Population size (<i>N</i>)	The average of structure scores		
	Asia	Car	Alarm
5	-2383.67	-5352.12	-10,298.6
10	-2378.28	-5332.56	-10,231.9
15	-2376.87	-5312.63	-10,183.6
20	-2375.40	-5312.55	-10,202.1
25	-2375.86	-5309.71	-10,161.5
30	-2375.42	-5307.30	-10,124.0
35	-2375.66	-5310.16	-10,087.8
40	-2375.57	-5309.88	-10,077.2

$$\frac{\text{Total number of ME edges}}{\text{Total number of edges}} \quad (6)$$

$$\frac{\text{Total number of EE edges}}{\text{Total number of edges}} \quad (7)$$

In the above relations, TC is the number of edges correctly added between the same nodes as those in target network. ME is the number of missed edges against target network. IE is the number of edges that have been determined correctly but are reverse and EE is the number of edges which are wrongly added.

To assess more accurately, the proposed method is run for each network 30 time individually and the average of above relations are listed in Table 4.

These tests highlighted that the resulting networks are close to the target networks and can conclude that the proposed method has suitable performance.

• Comparison with other methods

The proposed method has been compared with totally thirteen well-known methods. It has been tried to select the methods which are applicable and cover all different classes for Bayesian structure learning based on

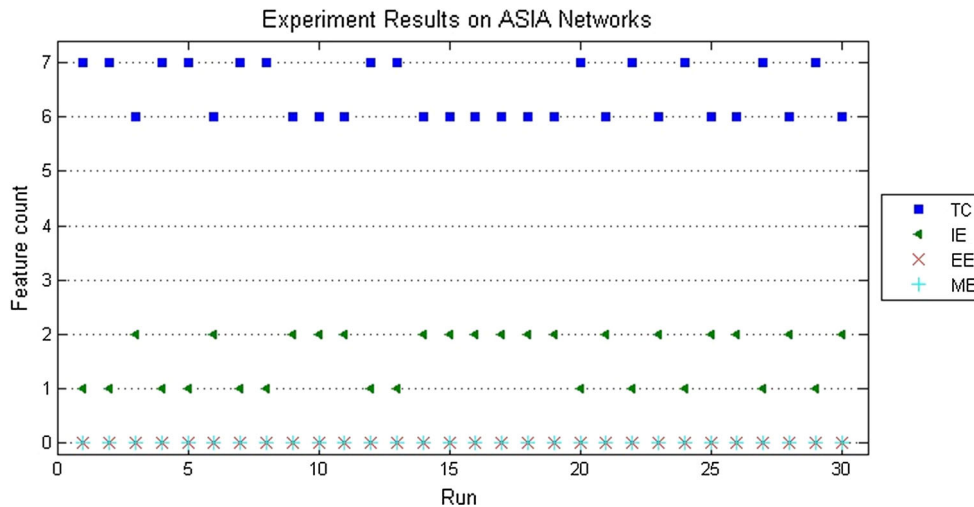


Fig. 9 Analysis the efficiency of the proposed method for ASIA Bayesian network

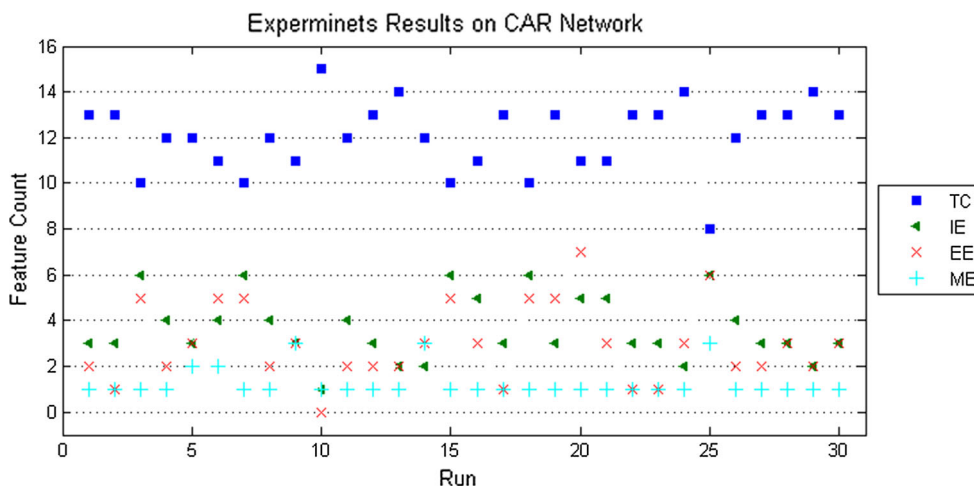


Fig. 10 Analysis the efficiency of the proposed method for Car Bayesian network

Fig. 11 Analysis the efficiency of the proposed method for Alarm Bayesian network

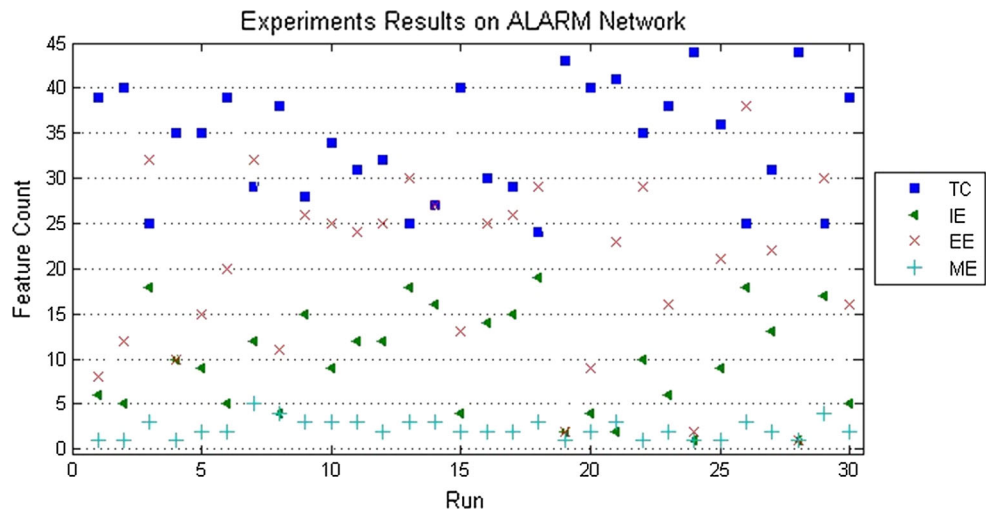


Table 4 The average of correct, missed, reverse and wrong edges over 30 times executions

Standard Bayesian network	TC%	IE%	ME%	EE%
Asia	80	20	0	0
Car	60	18	6	15
Child	57	8	5	30
Mildew	37	19	4	40
Insurance	44	12	2	42
Alarm	52	15	3	30
Hailfinder	50	2	5	43

data. These methods are REST (Cowie et al. 2007), CONAR (Cowie et al. 2007), CGA (Larrañaga 1996), SGA (You 2001), FGS (Chickering 2002), FCI (Colombo and Maathuis 2014), PC (Spirtes et al. 2000), MMHC (Tsamardinos et al. 2006), TPDA (Cheng et al. 1997),

SCA (Friedman et al. 1999), Minimum spanning tree (MST) (BayesiaLab 6.0.2), EQ (Munteanu and Bendou 2001) and Taboo (BayesiaLab 6.0.2).

Three evaluation metrics which are structure learning factor (SLF), topology learning factor (TLF) and graph error criteria have been used in which the first two metrics have been declared in relation 3.

Furthermore, graph error is the number of all errors such as reversing, missing and extra edges.

As mentioned above, among the comparing methods, REST and CONAR are based on the particle swarm optimization and CGA and SGA use the genetic algorithm. Since the proposed method has a hybrid strategy combining GA and PSO, for more assessment, the comparison of these four methods has been represented individually. The diagram in Fig. 12 shows the results for Asia, Car diagnosis and Alarm networks. Because of the stochastic behaviors of GA and PSO, the evaluating metrics have

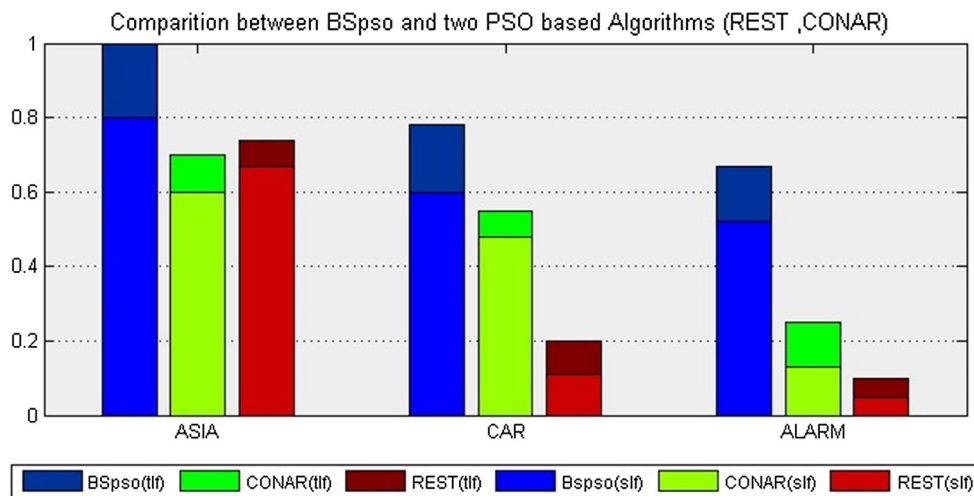


Fig. 12 Comparison the efficiency of the proposed method with REST and CONAR methods

Fig. 13 Comparison the normalized graph errors of our method with CGA and SGA methods

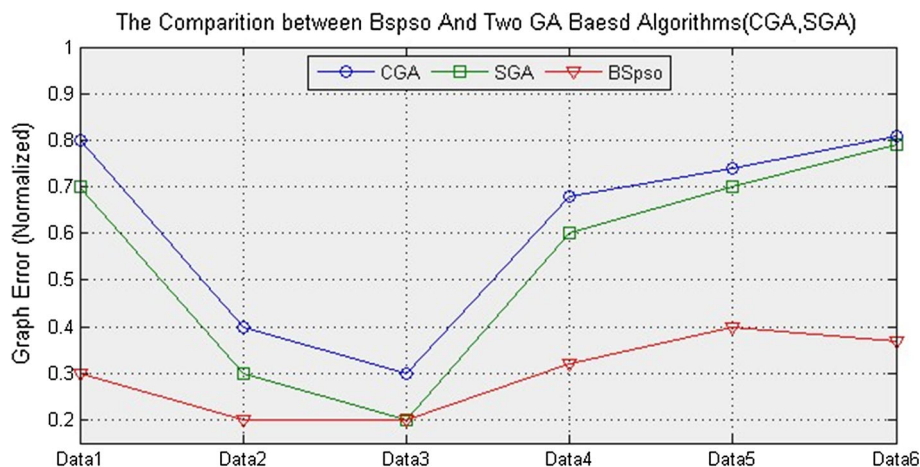


Table 5 Comparison of SLF measure among the proposed method and other nine algorithms on seven datasets

	FGS	FCI	PC	MMHC	TPDA	SCA	MST	EQ	Taboo	BSPSO
Asia	0.500	0.500	0.500	0.375	0.125	0.125	0.200	0.700	0.700	0.800
Car	0.529	0.176	0.183	0.059	0.059	0.471	0.294	0.647	0.647	0.605
Child	0.120	0.240	0.080	0.400	0.080	0.320	0.560	0.680	0.680	0.800
Alarm	0.804	0.217	0.565	0.652	0.629	0.435	0.261	0.674	0.717	0.956
Mildew	0.282	0.043	0.522	0.130	0.044	0.423	0.130	0.217	0.217	0.609
Insurance	0.462	0.211	0.115	0.327	0.135	0.192	0.250	0.500	0.423	0.769
Hailfinder	0.655	0.112	0.046	0.530	0.030	0.303	0.363	0.803	0.803	0.765

been calculated based on average over 30 runs. The results of SLFs and TLFs show that performance of the proposed method is higher than the other two methods. Also, in Alarm Bayesian network which is a challenging problem, the results of presented method are acceptable. This result has further strengthened our hypothesis that the combination of GA with PSO increases the chance of gaining optimal solution and prevents from early convergence to local optima.

In order to compare the proposed method with GA-based approaches (CGA and SGA), graph error criteria is used.

The diagram of comparing normalized graph errors of GA based and the proposed method are represented in Fig. 13.

The first three datasets which are related to ASIA network include 2000, 3000 and 5000 records, respectively. The second group of datasets, i.e., the fourth to sixth are related to Alarm network and include 3000, 5000 and 10,000 records, respectively. As expected, this experiment demonstrates that the graph errors of the proposed method are less than GA-based methods.

Next, the other nine methods have been compared according to the pervious assessment metrics and the results have been demonstrated separately in Tables 5, 6 and 7. In order to have a complete report, seven datasets are generated from the introduced benchmark networks. PC, SCA, MMHC and

TPDA has been run by using Causal Explorer system (Statinikov 2010). For FCI and FGS algorithms TETRAD project (Scheines 1998) has been used. Finally, BayesiaLab software (BayesiaLab 6.0.2) has been used for running Taboo, Minimum Spanning Tree and EQ methods. It is be noted that parameters of all the algorithms have been set based on their default values. In each row of Tables, the best value for each metric is shown in bold. Finally, we have reported the best results of proposed method over 30 runs.

SLF and TLF measures in Tables 5 and 6 demonstrate that the proposed method almost outperforms the other approaches and achieves the highest SLF on five datasets (asia, child, alarm, mildew and insurance) and obtains the highest TLF measures on five datasets (asia, car, child, mildew and insurance).

However, there are no major differences among BSPSO method and other approaches in other cases.

By analyzing the graph error measures in Table 7, it can be obtained that EQ and Taboo have smaller graph error measures than the other approaches in most cases. Maybe the main reasons for obtaining their efficient performance are that the first switches from greedy strategy to Taboo for avoiding from local minima and the second reduces the search space by searching for the equivalence classes of Bayesian networks. BSPSO achieves the best graph error score on four datasets (Asia, Car, Alarm and Hailfinder). Also its results

Table 6 Comparison of TLF measure among the proposed method and other nine algorithms on seven datasets

	FGS	FCI	PC	MMHC	TPDA	SCA	MST	EQ	Taboo	BSPSO
Asia	0.500	0.625	0.500	0.625	0.375	0.500	0.700	0.800	0.800	1
Car	0.529	0.294	0.235	0.059	0.118	0.588	0.647	0.647	0.647	0.772
Child	0.120	0.280	0.640	0.560	0.480	0.56	0.720	0.840	0.840	0.920
Alarm	0.891	0.217	0.565	1	0.629	0.674	0.674	0.891	0.891	0.956
Mildew	0.369	0.043	0.826	0.457	0.044	0.478	0.413	0.413	0.413	0.935
Insurance	0.558	0.211	0.558	0.557	0.519	0.346	0.442	0.615	0.577	0.962
Hailfinder	0.697	0.136	0.121	0.561	0.121	0.409	0.712	0.848	0.833	0.833

Table 7 Comparison of graph error among the proposed method and other nine algorithms on seven datasets

	FGS	FCI	PC	MMHC	TPDA	SCA	MST	EQ	Taboo	BSPSO
Asia	5	7	8	11	24	19	12	6	6	2
Car	8	19	17	18	35	35	8	3	3	9
Child	22	22	25	32	39	40	38	23	22	15
Alarm	19	36	60	50	60	65	38	12	9	2
Mildew	23	46	53	86	66	61	44	40	40	25
Insurance	39	42	58	64	61	79	12	9	9	18
Hailfinder	47	85	96	77	90	109	50	18	20	18

are affordable in other cases. Some methods such as PC, MMHC and SCA have poor outcomes. Maybe setting their parameters more accurately for more complex networks and increasing the number of samples in datasets improve their performances.

6 Conclusion

In this paper, Breeding Swarm PSO (BSPSO) algorithm as an efficient swarm intelligence-based approach has been used to solve the BN structure learning problem. BSPSO combines PSO and GA which provides a synergy between their capabilities and help to search more effectively the solution space and avoid from early convergence.

In order to set some parameters such as breeding rate, the method is run several times with different conditions. So as to assess the performance of the proposed method, seven datasets have been generated based on seven real-world networks. Several experiments have been done to evaluate the performance of the proposed method. Moreover, it has been compared with thirteen representative algorithms which four of them are GA based and PSO based, respectively. The experimental results demonstrate that BSPSO completely outperforms the four population-based approaches and also has promising performance against the other nine methods in finding the desirable networks.

Future work will concentrate on more complex problems in learning BNs, e.g., hidden variables and multi-relational data.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Abramson B et al (1996) Hailfinder: a Bayesian system for forecasting severe weather. *Int J Forecast* 12:57–71
- Ahmad FK, Deris S, Othman N (2012) The inference of breast cancer metastasis through gene regulatory networks. *J Biomed Inform* 45:350–362
- Alonso-Barba JI, Puerta JM (2011) Structural learning of Bayesian networks using local algorithms based on the space of orderings. *Soft Comput* 15:1881–1895
- BayesiaLab 6.0.2, Bayesia SAS, Laval, France. <http://www.bayesia.com>
- Beinlich IA (1989) The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks. Springer, Berlin
- Binder J et al (1997) Adaptive probabilistic networks with hidden variables. *Mach Learn* 29:213–244
- Cheng J, Bell DA and Liu W (1997) Learning belief networks from data: an information theory based approach. In: *Proceedings of the sixth international conference on Information and knowledge management*. ACM, pp 325–331
- Chickering DM, Geiger D, Heckerman D (1995) Learning bayesian networks: search methods and experimental results. In: *5th international workshop on artificial intelligence and statistics*, pp 112–128
- Chickering DM (1996) Learning Bayesian networks is NP-complete. In: Fisher D, Lenz H-J (eds) *Learning from data*, vol 112. Springer, New York, pp 121–130
- Chickering DM (2002) Optimal structure identification with greedy search. *J Mach Learn Res* 3:507–554
- Colace F, De Santo M, Vento M (2010) A multiexpert approach for Bayesian network structural learning. In: *43rd Hawaii interna-*

- tional conference on system sciences (HICSS), 2010. IEEE pp 1–11
- Colombo D, Maathuis MH (2014) Order-independent constraint-based causal structure learning. *J Mach Learn Res* 15:3741–3782
- Cowell R (1998) Introduction to inference for Bayesian networks. In: Jordan MI (ed) *Learning in graphical models*, vol 89. Springer, Netherlands, pp 9–26
- Cowie J, Oteniya L, Coles R (2007) Particle swarm optimisation for learning Bayesian networks. *World Congress on Engineering, Newswood Limited/International Association of Engineers (IAENG)*, pp 71–76
- Da You L et al (2001) Research on learning bayesian network structure based on genetic algorithms. *J Comput Res Dev* 8:916–922 (in Chinese)
- Friedman N, Nachman I, Peér D (1999) Learning Bayesian network structure from massive datasets: the “sparse candidate” algorithm. In: *Proceedings of the fifteenth conference on uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp 206–215
- Gómez JA, Mateo JL, Puerta JM (2011) Learning Bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. *Data Min Knowl Discov* 22:106–148
- Heckerman D, Geiger D, Chickering DM (1995) Learning Bayesian networks: the combination of knowledge and statistical data. *Mach Learn* 20:197–243
- Hill SM et al (2012) Bayesian inference of signaling network topology in a cancer cell line. *Bioinformatics* 28:2804–2810
- Jensen AL, Jensen FV (1996) MIDAS—an influence diagram for management of mildew in winter wheat. In: *Proceedings of the twelfth international conference on uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp 349–356
- Ji J et al (2011) A hybrid method for learning Bayesian networks based on ant colony optimization. *Appl Soft Comput* 11:3373–3384
- Ji J, Wei H, Liu C (2013) An artificial bee colony algorithm for learning Bayesian networks. *Soft Comput* 17:983–994
- Khanteymooiri AR, Menhaj MB, Homayounpour MM (2011) Structure learning in Bayesian networks using asexual reproduction optimization. *ETRI J* 33:39–49
- Larrañaga P et al (2013) A review on evolutionary algorithms in Bayesian network learning and inference tasks. *Inf Sci* 233:109–125
- Larrañaga P et al (1996) Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters. *IEEE Trans Pattern Anal Mach Intell* 18:912–926
- Lauritzen SL, Spiegelhalter DJ (1988) Local computations with probabilities on graphical structures and their application to expert systems. *J R Stat Soc B* 50:157–224
- Li Z et al (2011) Large-scale dynamic gene regulatory network inference combining differential equation models with local dynamic Bayesian network analysis. *Bioinformatics* 27:2686–2691
- Mattew S, Terence S (2006) Breeding PSO: a GA/PSO Hybrid. Department of Computer Science, University of Idaho, Moscow
- Munteanu P, Bendou M (2001) The EQ framework for learning equivalence classes of Bayesian networks. In: *Proceedings iee international conference on data mining, 2001. ICDM 2001. IEEE*, pp 417–424
- Murphy K, Mian S (1999) Modelling gene expression data using dynamic Bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, CA
- NorsysSoftwareCorp, 1990–2013. Netica. Version 5.12. <http://www.norsys.com/>
- Robinson RW (1977) Counting unlabeled acyclic digraphs. In: Little CHC (eds) *Combinatorial mathematics V. Lecture Notes in Mathematics*, vol 622. Springer, Heidelberg, pp 28–43
- Scheines R et al (1998) The TETRAD project: constraint based aids to causal model specification. *Multivar Behav Res* 33:65–117
- Spiegelhalter DJ, Cowell RG (1992) Learning in probabilistic expert systems. *Bayesian Stat* 4:447–465
- Spirtes P, Glymour CN, Scheines R (2000) *Causation, prediction, and search*. MIT press, Cambridge
- Statnikov A (2010) Causal explorer: a matlab library of algorithms for causal discovery and variable selection for classification. *Causation Predict Chall Chall Mach Learn* 2:267
- Tsamardinos I, Brown LE, Aliferis CF (2006) The max–min hill-climbing Bayesian network structure learning algorithm. *Mach Learn* 65:31–78
- Wong ML, Lam W, Leung KS (1999) Using evolutionary programming and minimum description length principle for data mining of Bayesian networks. In: *IEEE transactions on pattern analysis and machine intelligence*, pp 174–178
- Yang G, Lin Y, Bhattacharya P (2010) A driver fatigue recognition model based on information fusion and dynamic Bayesian network. *Inf Sci* 180:1942–1954
- Ziegler V (2008) Approximation algorithms for restricted Bayesian network structures. *Inf Process Lett* 108:60–63
- Zou M, Conzen SD (2005) A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics* 21:71–79