Swarm and Evolutionary Computation 76 (2023) 101224

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

An efficient Bayesian network structure learning algorithm based on structural information

Wei Fang [a,*], Weijian Zhang [a], Li Ma [a], Yunlin Wu [a], Kefei Yan [a], Hengyang Lu [a], Jun Sun [a], Xiaojun Wu [a], Bo Yuan [b]

[a] International Joint Laboratory on Artificial Intelligence of Jiangsu Province, Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi, Jiangsu, China

[b] Computer Science and Engineering Department, Southern University of Science and Technology, Shenzhen, China

ARTICLE INFO

Keywords:
Bayesian networks
Genetic algorithm
Markov blanket
v-structure
Structure learning

ABSTRACT

Bayesian networks (BNs) are probabilistic graphical models regarded as some of the most compelling theoretical models in the field of representation and reasoning under uncertainty. The search space of the model structure grows super-exponentially as the number of variables increases, which makes BN structure learning an NP-hard problem. Evolutionary algorithm-based BN structure learning algorithms perform better than traditional methods. This paper proposes a structural information-based genetic algorithm for BN structure learning (SIGA-BN) by employing the concepts of Markov blankets (MBs) and v-structures in BNs. In SIGA-BN, an elite learning strategy based on an MB is designed, allowing elite individuals' structural information to be learned more effectively and improving the convergence speed with high accuracy. Then, a v-structure-based adaptive preference mutation operator is introduced in SIGA-BN to reduce the redundancy of the search process by identifying changes in the v-structure. Furthermore, an adaptive mutation probability mechanism based on stagnation iterations is adopted and used to balance exploration and exploitation. Experimental results on eight widely used benchmark networks show that the proposed algorithm outperforms other GA-based and traditional BN structure learning algorithms regarding structural accuracy, convergence speed, and computational time.

1. Introduction

Bayesian networks (BNs), as methods of reasoning under uncertainty, are some of the best approaches for representing causal knowledge and are very popular in the field of probability [1]. A directed acyclic graph (DAG), in which nodes represent random variables, and the existence of arcs denotes the dependence relationships between variables, is typically used to represent BNs. These relationships are quantified by a set of conditional probability distributions (CPDs) determined by the parent nodes of each variable. Due to the advantages of BNs regarding their inference and learning abilities, these networks have become increasingly popular in various research areas, such as bioinformatics research [2], medical problems [3], and image processing [4]. Before using BN for analysis and reasoning, obtaining the network model with structure and parameter learning is necessary. Structure learning is the basis of parameter learning and the premise of applying BN. The structure of a BN can be provided manually by experts, but the accuracy cannot be guaranteed, and this process is also time-consuming. Therefore, learning a BN structure from data is an important task studied extensively during the last two decades.

BNs consist of DAG and conditional probability tables (CPT), which are used to describe the structure and parameters of BNs, respectively. Learning a BN structure means learning the topology of the network. Learning a completely correct BN structure from data is an NP-hard problem when the number of variables increases rapidly [5]. Several algorithms have been proposed to solve BN structure learning problems and approximately learn BN structures. There are three commonly used methods to learn BNs structure from data: constraint-based approaches, scored-based approaches, and hybrid approaches.

Constraint-based approaches first identify conditional independence (CI) relations between variables through statistical methods such as Pearson's chi-square test [6]. Then, the BN structure that best fits those relations is constructed. In 1990, Spirtes [7] et al. proposed the first CB method: the SGS algorithm. Some widely known algorithms are the PC algorithm [8], grow–shrink algorithm (GS) [9], tree-augmented naive

[*] Corresponding author.
E-mail address: fangwei@jiangnan.edu.cn (W. Fang).

https://doi.org/10.1016/j.swevo.2022.101224
Received 27 April 2022; Received in revised form 25 October 2022; Accepted 25 November 2022
Available online 3 December 2022

Bayes (TAN) algorithm [10], maximum weight spanning tree (MWST) algorithm [11], etc.

Score-based algorithms evaluate the quality of candidate network structures with a scoring metric. These approaches, such as the K2 algorithm [12], regard the BN structure learning problem as a combinatorial optimization problem. Traditional single-solution search algorithms, such as the hill climbing algorithm (HC) [13] and simulated annealing, may be trapped in local optima [1]. Some population-based search algorithms, such as the genetic algorithm (GA) [14–16], particle swarm optimization (PSO) [17,18], and ant colony optimization (ACO) [19], have been introduced to identify optimal structures. Some of the best-known score-based algorithms include the K2 search GA (K2GA) [20], Chain-model GA [21], etc. The above two algorithms have their advantages and disadvantages. The constraint-based approach is more sensitive to noise in the data, and the accuracy of the high-order CI test cannot be guaranteed. The Score-based algorithms will lead to low search efficiency due to the huge search space.

Hybrid algorithms integrate score-based and constraint-based approaches to search BN structures in a large search space [1]. One commonly used strategy is to employ constraint-based approaches to construct a graph's skeleton and then utilize a score-based approach to search for the optimal DAG with the highest score. The max–min hill climbing (MMHC) [22] and sparse-candidate (SC) algorithms [23] are the representative hybrid algorithms. In recent years, Constantinou proposed the SaiyanH [24] algorithm using a mixture of CB and SS search strategies in three stages. Evolutionary Algorithms (EAs), such as simulated annealing (SA) and Genetic Algorithm (GA), have also been proposed to solve the BN structure learning problem [14,15,15,16,25,26]. However, GA-based methods have difficulty using elite individual information and contain redundant searches, requiring a long execution time to achieve better results. This paper proposes a structure information-based GA for BN structure learning (SIGA-BN) to make the searching process of GA more effective. We combine BN structure information with evolution operators to design an effective algorithm. The main contributions are summarized as follows.

1. An elite learning strategy using Markov blanket (MB) information is designed to use elite individuals better. With the proposed learning strategy, an individual can learn local characteristics from elite individuals more efficiently. In addition, the convergence speed can be accelerated by learning the information of the MB.
2. A v-structure-based adaptive preference mutation operator is proposed to make the search process more efficient. The v-structure concept is introduced in the mutation operator, which prefers larger v-structure variations, to reduce the probability of redundant searches. An adaptive mutation probability adjustment mechanism based on stagnant iterations is also designed to adjust preferences and enhance exploitation. The proposed mutation operator strongly helps the proposed algorithm find a BN with higher structural accuracy in less execution time.

The remainder of this paper is organized as follows. Section 2 provides a brief review of the related work on BN structure learning algorithms. Section 3 introduces BNs and commonly used approaches to learn BN structures with GAs. In Section 4, SIGA-BN is briefly presented. The benchmark problems, the compared algorithms, and experimental results with analyses are presented in Section 5. Finally, conclusions are drawn in Section 6.

2. Related work

As mentioned in Section 1, constraint-based approaches, score-based approaches, and hybrid approaches are the main types of BN structure learning algorithms.

As the earliest BN structure learning algorithms, constraint-based approaches usually constructs a BN according to a set of CI relations between variables estimated by statistical methods, such as Pearson's chi-square test [6]. The PC algorithm [8] starts with a complete undirected graph and recursively deletes the edges according to the CI test. Part of the undirected graph is oriented and further extended to represent a DAG. When the DAG is a sparse network, the PC algorithm can reduce the required computations and adopt the chi-square test or mutual information test without needing a specific CI test. The Gerchberg–Saxton(GS) algorithm [9] is a two-stage learning algorithm, which is growing stage and shrink stage, according to the concept of MB. In the growth stage, as many potential nodes are added as possible, while redundant nodes are rigorously detected and eliminated in the shrinkage stage. The three-phase dependency analysis (TPDA) algorithm [27] builds BNs through a three-stage algorithm: drafting, thickening, and thinning. The TPDA algorithm can deal with the BN structure learning problem in the case of known node orders and possesses better time performance and accuracy than the PC algorithm. The IAMB algorithm proposed by Tsamardinos et al. in 2003 and the subsequent optimization based on it, such as Inter-IAMB [28], learn structures by learning node-dependent MBs. In 2019, Qi et al. improved the PC algorithm and proposed an algorithm called PC-MI [29].

Scoring metrics can be used to evaluate the quality of a structure, so these approaches regard the BN structure learning problem as a combinatorial optimization problem. The K2 algorithm [12], proposed in 1992, assumes that the order of variables is available and all structures are equally possible. The K2 algorithm searches each node and uses a greedy heuristic method to search for a set of parent nodes that maximizes the score. The K2 algorithm does not always find the optimal set of parents that maximizes the score but instead increases the number of parents until the scores of variables stop rising. Shahab Behjati et al. [30] made improvements to the K2 algorithm in 2020. The HC algorithm [13] searches the neighborhood and changes one edge during each iteration until the score is optimized. This kind of algorithm can converge quickly in cases with small numbers of nodes but has a low possibility of obtaining the global optimal solution. Therefore, several EA-based BN structure learning algorithms have been proposed. The BN structure learning with PSO (BNC-PSO) [1] algorithm combines PSO with the GA and includes two novel formulas for velocity and position updating. The binary encoding water cycle algorithm (BEWCA-BN) [17] was proposed to address the BN structure learning problem. In 2021, Xu et al. proposed a full permutation and extensible ordering-based search algorithm [31].

With the growth of the search space, score-based methods have difficulty obtaining better results in a limited timeframe. Therefore, hybrid approaches combine constraint-based and scored-based approaches to address this problem. Constraint-based approaches are commonly used for reducing the given search space, and score-based approaches can search for a better BN. The MMHC method [22] proposed by Tsamardinos et al. is a hybrid structure learning method. In the constraint-based phase, the multiple model-based predictive control (MMPC) method uses the CI test to determine each node's parent and child node sets to constrain the search space and determine a skeleton. In the score-based search stage, the HC algorithm determines the structure of the desired DAG. As a traditional EA, GA can find optimal solutions in a reasonable amount of time and is also widely applied in the BN structure learning field. The K2GA [20], as the first proposed GA-based hybrid method, evolves an appropriate order of nodes. The K2GA [20] regards the orders of nodes as individuals, and the K2 algorithm is a sub-process used to calculate each individual's fitness. GA is used to find an optimal order of nodes, transmitted to the K2 algorithm for constructing the final BN. The hybrid structure learner GA (HSL-GA) [14] was proposed to solve the BN structure learning problem in cases with limited datasets in large and medium-sized networks. In HSL-GA, mutual independence is detected to reduce the complexity of the search space, and GA is used to search the possible structures in the reduced search space effectively. [25] solves the problem of adaptive guidance parameters based on an elite population. In AESL-GA, a

Table 1
Summary of BN structure learning algorithms.

Algorithm classification	Algorithm abbreviation	Algorithm name or description	Year	Score	Search space	Exact algorithm
Constraint-based approaches	MWST [11]	Maximum weight spanning tree algorithm	1968	–	–	–
	SGS [7]	Spirtes-Glymour-Scheines	1990	–	–	–
	PC [8]	Peter and Clark	1991	–	–	–
	TAN [10]	Tree-augmented naive Bayes algorithm	1997	–	–	–
	TPDA [27]	Three Phase Dependency Algorithm	2002	–	–	–
	IAMB [28]	Incremental Association Markov Blanket	2003	–	–	–
	Inter-IAMB [28]	Interleaved-IAMB	2003	–	–	–
	MMPC [37]	Max–Min Parents Children	2003	–	–	–
	GS [9]	Gerchberg–Saxton algorithm	2008	–	–	–
	PC-MI [29]	A PC variant empowered by the WMIF strategy	2019	–	–	–
Score-based algorithm	K2 [12]	K2	1992	K2	DAG	NO
	HC [13]	Hill Climbing	1995	BDeu	DAG	NO
	BEWCA-BN [17]	Binary encoding water cycle algorithm	2018	k2	DAG	NO
	BNC-PSO [1]	BN structure learning with PSO algorithm	2016	AIC, BIC, DIC, JIC	DAG	NO
	ACO [19]	Ant colony optimization	2002	K2	DAG	NO
	Improved K2 algorithm [30]	Improved K2 algorithm	2020	BIC	DAG	NO
	PEWOBS [31]	Full permutation and extensible ordering-based search algorithm	2021	AP, Recall	DAG	YES
	Model selection for BNs via the bootstrap [38]	Model selection for BNs via the bootstrap	2021	BIC	DAG	YES
Hybrid approaches	MMHC [22]	Max–min Parents and Children and Hill Climbing	2006	BDeu	DAG	NO
	AESL-GA [25]	An elite-guided genetic algorithm	2018	BDeu	DAG	NO
	Hybird-SLA-GA [15]	Hybrid-Structure Learning Algorithm	2019	BDeu	DAG	NO
	EKGA-BN [26]	Effective Knowledge-driven GA	2020	BDeu	DAG	NO
	Local-DSLA [39]	Decomposition-based BN Structure Learning Algorithm using Local topology information	2020	BIC, BDeu	DAG	YES
	SaiyanH [24]	Saiyan Hybrid	2020	SHD, BF	DAG	NO
	ARCS [32]	Annealing on regularized Cholesky score	2021	SHD, JI	DAG	NO
	Improved PSO [40]	An improved Particle Swarm Optimization (PSO) algorithm	2021	HD, BIC	DAG	YES
	PC-PSO [33]	An algorithm combining PC and particle swarm optimization	2021	BIC	DAG	YES
	SC [23]	Sparse-candidate algorithms	1999	BDeu	DAG	YES
	K2GA [20]	K2 search GA	1996	–	DAG	NO
	HSL-GA [14]	Hybrid structure learner GA	2014	BDeu	DAG	NO
	DFA-B [34]	A discrete firefly optimization algorithm	2022	K2	DAG	YES

series of parameterless hybrid methods are designed and a knowledge-driven strategy is proposed to select the parent nodes and adjust the maximum parent threshold of each node. EKGA-BN [26] combines the HC algorithm with a selection operator and designs a novel knowledge-driven mutation schema. The hybrid-SLA-GA [15] uses an improved GA to search in the constrained search space and combines CHC algorithm with GA, which can get better results. In recent years, hybrid algorithms such as SaiyanH [24], ARCS [32], PC-PSO [33], and DFA-B [34] have been proposed. Table 1 summarizes the three types of BN structure learning algorithms. Two surveys are reviewed on the BN structure learning algorithms in [35,36].

3. Preliminaries

In this section, a brief summary of the BN structure learning problem and related BN concepts are introduced.

3.1. Problem statement

Let $G = (X, E)$ be a DAG where $X = \{X_1, X_2, \dots, X_n\}$ is a node set representing variables and $E = \{e_{ij}\}$ is a directed edges set representing the independence relationships between these variables. The relationships are represented by E, which is combined with directed arcs $e_{i,j}$ from parent node X_i to child node X_j. $Pa(X_i)$ is defined as the parent node set of X_i. The dependence relationship between X_i and X_j can be quantified with the CPD $P(X_i \| Pa(X_i))$. If (G, P) satisfies the Markov condition, (G, P) can be called a BN [17], where the joint probability distribution P is combined by a product of local CPDs according to (1):

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)). \tag{1}$$

Score-based approaches define a score metric to evaluate the matching rate between the network and the observed data, and then obtain a BN structure that obtains the highest score during the procedure.

The score metrics can be categorized into two classes called Bayesian and information-theoretic scoring functions. The Akaike information criterion (AIC) [41], Bayesian information criterion (BIC) [42], and minimum description length (MDL) [43] are well-known information-theoretic scoring metrics. For a Bayesian scoring metric, given a training dataset D, the general idea is to compute the posterior probability distribution of a graph G and penalize this score by the complexity of the BN according to (2)

$$p(G|D) = \frac{p(D|G)p(G)}{p(D)} \propto p(D|G)p(G), \tag{2}$$

where $p(G)$ denotes the prior probabilities on different graph structures and $p(D|G)$ is the parameter prior that computes probabilities on different parameters Θ given a graph G. According to Bayes theorem, $p(D|G) = \int_{\theta} p(D|G, \theta)p(\theta, G)d\theta$, which is the marginal likelihood that averages the probabilities of the data D over all possible parameter assignments to G [17].

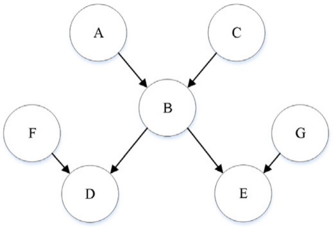


Fig. 1. Markov blanket.

3.2. Markov blanket

The MB of a node X_i in a BN denoted as MB(X_i), refers to the nodes consisting of X_i's children, X_i's parents, and other parents of X_i's children, which separates node X_i from the rest of the nodes in the BN [44]. As shown in Fig. 1, MB(B) = $\{A, C, D, E, F, G\}$. The MB of a node can be understood as a minimum node set that makes the node conditionally independent of the other nodes. The remaining nodes not in MB(X_i) are irrelevant to X_i and are regarded as redundant nodes of X_i. In other words, the relevant structure information of a node is contained in its MB. In this paper, we adopt the MB of a node to represent the characteristics of an individual, which means that MB(X_i) can be used to indicate the partial characteristics of a BN.

3.3. v-structure

Let nodes X and Z be the parents of node Y. Then, nodes X, Y, and Z form a v-structure in a DAG, denoted as $X \rightarrow Y \leftarrow Z$. Different BN structures of different BNs may actually contain the same independence assertions, meaning that different BN structures may be equivalent [44]. I-equivalence helps identify two BNs when learning BN structures from data since this approach can achieve independent equivalence between two BNs. The I-equivalence of a BN G, denoted as $I(G)$, can be inferred from its skeleton and v-structure, which can be used to estimate whether two BNs are equivalent. If BNs G_1 and G_2 have the same skeleton and v-structure, they are equivalent, that is, $I(G_1) = I(G_2)$, which is a sufficient condition for I-equivalence [44]. If two BNs are I-equivalent, they have the same score, also called score equivalence. Score-based approaches prefer to obtain a solution with a higher score in the search space as often as possible, and evolving equivalent solutions frequently reduces the search efficiency of the algorithm. Therefore, identifying the I-equivalence relations among BNs helps reduce the search process's redundancy. In Fig. 2, the three left networks have different structures, but they have the same CI relations with each other, which is $X \perp Y|Z$; therefore, they are I-equivalent. The rightmost network in Fig. 2 is not I-equivalent with the three left networks.

3.4. BN structure learning based on the GA

The use of a scoring metric to learn BN structures can be regarded as a combinatorial optimization problem, which the GA can solve with the following steps:

1. Encoding is completed for the BN structure learning problem with an adjacency matrix or a set of edges existing in a DAG. A BN structure with n variables can be represented by an $n \times n$ adjacency matrix. An individual can therefore be represented by the in string (3):

$$x_i = a_{11}a_{12}...a_{1n}a_{21}a_{22}...a_{ij}...a_{nn}, \tag{3}$$

where a_{ij} denotes whether node i is the parent of node j and can be calculated using (4):

$$a_{ij} = \begin{cases} 1 & if\ i\ is\ a\ parent\ of\ j \\ 0 & otherwise. \end{cases} \tag{4}$$

2. The initial population is constructed randomly with N individuals. Random initialization may generate invalid individuals in various cycles. Thus, it is necessary to remove cycles with few changes in the DAGs. Each individual is encoded as in (3), and the fitness can be calculated according to the predefined scoring metric.
3. A selection operator is used to retain the individuals with the highest fitness. For instance, roulette wheel selection and tournament selection are two widely used selection operators. The selection operator inserts an individual with a higher score into the new population, and this individual can be transferred to the next generation
4. The crossover operator and mutation operator are carried out for each individual to generate new individuals, which may also introduce cycles into the DAGs. Therefore, a cycle removal procedure is also needed after executing the crossover and mutation operators to make the obtained individuals valid.
5. For each individual in the next generation, steps 3 and 4 are repeated until the termination condition is satisfied and the output the individual with the highest score is regarded as the final solution.

4. Proposed algorithm

4.1. Motivation

Social learning refers to learning behaviors from better individuals in a population, which is common among social animals. Learning rates can be accelerated based on social learning [45–47]. In this paper, we introduce the mechanism of social learning in the proposed algorithm based on an MB to learn good characteristics from elite individuals more effectively. The other individuals learn the structural information in the MB of any elite individual, which improves the convergence speed with high accuracy.

As introduced in Section 3.3, BNs with the same skeleton and v-structure are equivalent, meaning that different corresponding individuals have the same performance. It is not necessary to search the equivalent individuals in the search space. In this paper, we introduce the v-structure concept to design a novel mutation operator that makes individuals prefer to have significant v-structure changes after the mutation operator is executed, thereby enhancing the effectiveness of the search process. At the same time, a fixed mutation probability may make the algorithm fall into local optima. Therefore, we propose a mechanism to modify the mutation probability of preference based on the number of stagnation iterations, which balances the exploration and exploitation tendencies of the algorithm.

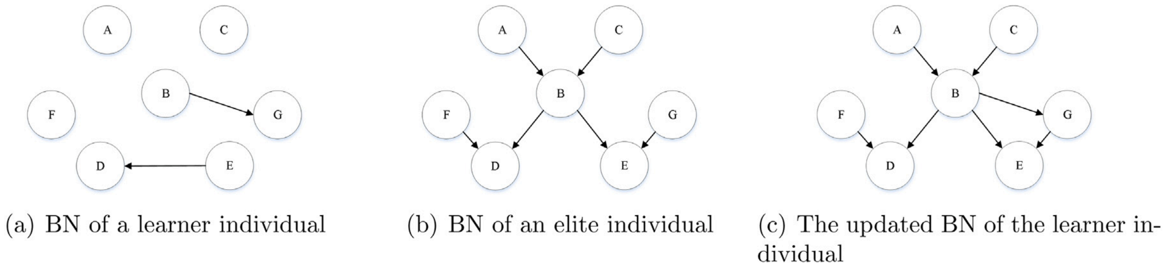
Based on the above two motivations, this paper proposes the SIGA-BN algorithm, which can better use the structural information of BNs.

4.2. Elite learning strategy based on an MB

In the proposed SIGA-BN, an elite set S_{elite} is constructed with the best N_{elite} individuals according to the fitness values obtained after the execution of the selection operator. A uniform crossover operator is then carried out in the population. The elite set S_{elite} is used as the demonstrators to guide the evolution process more efficiently. In the mechanism of social learning, each individual's task of learning from better individuals may differ. Hence, a learning probability is defined for each individual. Each individual learns from any randomly selected elite individual in S_{elite} only if a randomly generated probability is larger than the predefined learning probability. The designed learning strategy for the individual is to copy the structural information of MB of a node from the randomly selected elite individual to its structure. Based on the learning strategy, good structure information among elite individuals can be learned to accelerate the evolution process.

Z
X
Y
Z
Y
X
Z
X
Y
Z
X
Y

Fig. 2. Three different structures with the same I-equivalence.



(a) BN of a learner individual

(b) BN of an elite individual

(c) The updated BN of the learner individual

Fig. 3. Example of the proposed elite learning strategy based on an MB.

Fig. 3 shows an example of the proposed learning mechanism based on an MB. Fig. 3(a) gives the BN structure of a learner individual, in which B is the chosen node that needs to learn structural information from an elite individual. Fig. 3(b) shows the BN structure of a randomly selected elite individual. The learner individual learns the local structure information of node $B's$ MB from the elite individual. After the learning process, the BN structure of the learner individual is updated, as shown in Fig. 3(c).

The proposed elite learning strategy based on an MB is illustrated in Algorithm 1.

Algorithm 1: Elite learning strategy based on MB

```
Require: Population before crossover, $P_{prev}$;
    Population after crossover, $P_c$;
    Elite learning probability, $P_{learn}$;
    Population size of $P_{prev}$ and $P_c$, $N_{prev}$ and $N_c$;
    Elite individual set size, $N_{elite}$;
Ensure: Population after the learning strategy, $P_c$;
Obtain the top $N_{elite}$ scoring individuals in $P_{prev}$ as the elite set $S_{elite}$.
for i=1 to $N_c$ do
  if $rand() < P_{learn}$ then
    Choose one individual $G_e$ in the elite set $S_{elite}$ randomly.
    Choose one node $n_0$ in $G$ randomly.
    Obtain the parent nodes $Pa(n)$ and child nodes $Ch(n)$ of node $n_0$ from $G_e$.
    Replace the parent nodes of node $n_0$ in the original individual $G$ with $Pa(n)$.
    for $n_{ch}$ in $Ch(n)$ do
      Get parent nodes $Pa(n)$ of node $n_{ch}$ from $G_e$.
      Replace parent nodes of node $n_{ch}$ in origin individual $G$ with $Pa(n)$.
    end for
  end if
end for
```

4.3. V-structure-based adaptive mutation operator

Since BNs with the same skeleton and v-structure are equivalent, an individual may be unchanged after the mutation operator, resulting in low search efficiency. In the proposed BN structure learning algorithm based on the GA, we prefer the mutation results that yield significant changes in the structure, which can be checked according to the v-structure. For an edge in a BN such as $A \leftarrow B$, there are two kinds of mutation methods: $A \rightarrow B$ or $A \nleftrightarrow B$. Fig. 4 gives an example of two possible mutation results. Fig. 4(a) shows a BN with four nodes. Figs. 4(b) and 4(c) are two possible mutation results based on the BN in Fig. 4(a). In Fig. 4(b), only the v-structure of node B has been changed, and the number of changes is one. In Fig. 4(c), the v-structure of node B has been changed and the v-structure of node A is got, and then the number of changes is two. Therefore, the mutation results in Fig. 4(c) can be accepted with higher probability. In this paper, the proposed mutation operator is carried out according to the two different mutation methods conducted on the selected edge in a BN individually. The mutation result with a greater number of changes in the v-structure remains in the next generation, which helps to find the non-equivalent BN structures and guides the search process efficiently.

To address the local convergence problem in the GA, an adaptive preference probability-changing mechanism is proposed here by taking the number of stagnation iterations into account with the following equation:

$$P_{prefer} = (e^{-\frac{NoStag}{4}} + 1)/2 \tag{5}$$

where P_{prefer} is the preference probability, and $NoStag$ denotes the number of stagnation. As seen from (5), the preference probability decreases slowly from 1 as the number of stagnation iterations increases; this can yield a good balance between exploration and exploitation.

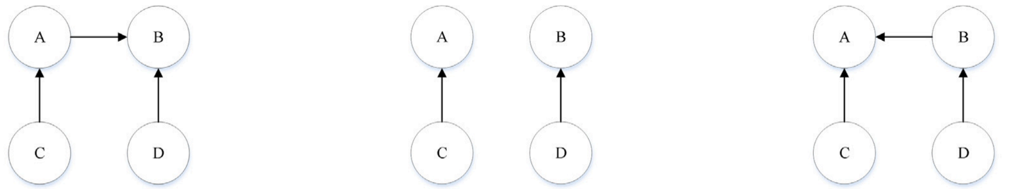
In general, the proposed v-structure-based adaptive mutation operator prefers to focus on the nonequivalent BNs as much as possible and then reduces the redundancy of the search process, which increases the probability of the algorithm finding a better BN.

The details of the proposed v-structure-based adaptive mutation operator are shown in Algorithm 2.

4.4. The proposed algorithm

The procedure of SIGA-BN is shown in Fig. 5. The main steps are summarized as follows.

- Step 1: Mutual dependencies. Statistical tests are used to build an undirected graph structure, which is referred to as a superstructure (SS) and helps to reduce the search space [14]. If node i and node j are conditionally independent, the undirected edge e_{ij} is added to $SS = \{e_{ij}\}$. Each edge in SS has 3 states: $A \leftarrow B$, $A \rightarrow B$, and $A \nleftrightarrow B$.
- Step 2: Random initialization. For each undirected edge in the SS, one of the states is randomly chosen to initialize the population.



(a) Original BN (b) Mutation result 1 (c) Mutation result 2

Fig. 4. Example of two possible mutation results for a BN.

Algorithm 2: v-structure-based adaptive mutation operator

```
Require: Individual, $G$;
    The number of stagnation iterations, $s$;
Ensure: Individual after improved mutation, $G_{new}$;
Adaptive preference rate $Pr = \frac{e^{-\frac{s}{4}}+1}{2}$
Get the edge set E of the individual G
for i=1 to size(E) do
    if $rand < P_{prefer}$ then
        For possible mutation type 1 $m_1$ and type 2 $m_2$ of the
        edge E(i), calculate the number of possible changes with
        the v-structure as $c_1$ and $c_2$, respectively.
        if $c_1 = c_2$ then
            Choose the mutation type randomly and apply it to $G$.
        else
            if $c_1 > c_2$ then
                if $rand < P_{prefer}$ then
                    choose $m_1$
                else
                    choose $m_2$
                end if
            else
                if $rand < P_{prefer}$ then
                    choose $m_2$
                else
                    choose $m_1$
                end if
            end if
        end if
    end if
end for
```

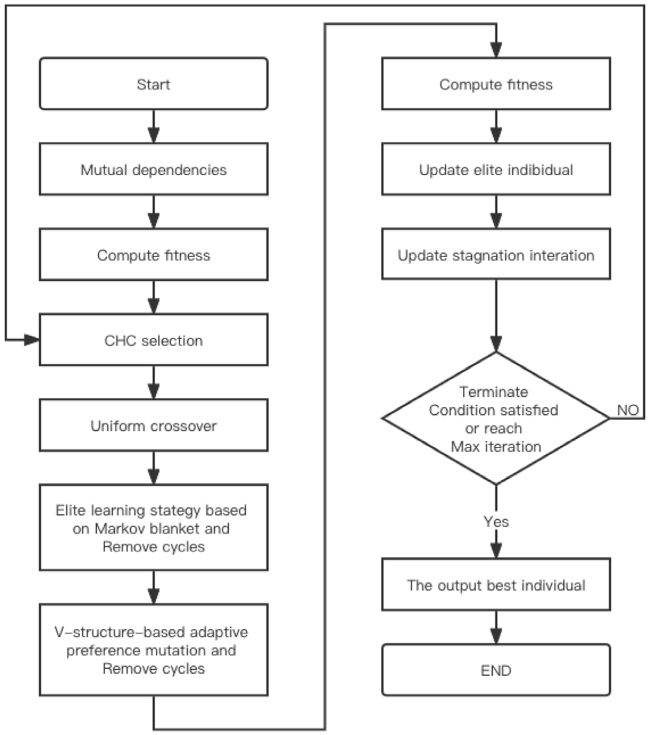

Start
Mutual dependencies
Compute fitness
CHC selection
Uniform crossover
Elite learning stategy based on Markov blanket and Remove cycles
V-structure-based adaptive preference mutation and Remove cycles
Compute fitness
Update elite indibidual
Update stagnation interation
Terminate Condition satisfied or reach Max iteration
NO
Yes
The output best individual
END

Fig. 5. Procedure of SIGA-BN.

- Step 3: Cycle removal. During the initialization, crossover, and mutation procedures, cycles may be introduced. Nodes with several parent nodes larger than the predefined N_{mp} might be produced, where N_{mp} denotes the maximum number of parent nodes for each node. Consequently, a procedure to remove cycles in invalid individuals is necessary. The cycle removal procedure randomly drops redundant parent nodes at first and then solves the minimum cycles set problem with an improved GR algorithm [14]. The removing cycles algorithm prefers to remove edges between nodes with more children and fewer parents to minimize the number of changes.
- Step 4: Fitness evaluation. The uniform joint distribution likelihood equivalence Bayesian Dirichlet (BDeu) score is used to evaluate each individual's fitness after the initialization, crossover, and mutation procedures at each iteration.
- Step 5: Selection operator. The CHC selection operator [15] is used to choose the individuals from the current population and the offspring population.
- Step 6: Crossover operator and elite learning strategy based on an MB. Before uniform crossover is performed, an elite set is generated with the best N_{elite} individuals. Then, one of each individual's nodes learns the MB's structural information from the randomly selected elite individual with the preset learning probability. Therefore, the individuals contain information from both their parents and the elite individuals, thus accelerating the convergence rate.
- Step 7: V-structure-based adaptive mutation operator. The preference probability P_{prefer} is calculated according to (5). For each muted gene site of an individual, the number of v-structure changes is calculated. The mutation result with the greatest number of v-structure changes is selected with a probability Pm. The proposed mutation strategy prefers to focus on the non-equivalent BNs as much as possible, reducing the redundant search process.
- Step 8: Stagnation iteration update. If the best score in the current population is not better than the score of the previous generation, the number of stagnation iterations is increased linearly. Otherwise, the number of stagnation iterations is reset to 1. The adaptive preference probability-changing approach can balance the algorithm's exploration and exploitation tendencies.

Step 3 to Step 8 are repeated until the termination condition is satisfied, and then the final solution with the highest BDeu score is selected as the identified BN structure.

Table 2
Datasets used in experiments.

Datasets	Original network	Number of cases	Number of nodes	Number of arcs	Score
Asia-500	Asia	500	8	8	−1158.323
Sachs-500	Sachs	500	11	17	−3895.191
Insurance-500	Insurance	500	27	52	−8213.343
Alarm-500	Alarm	500	37	46	−5176.457
Barley-500	Barley	500	48	84	−40 278.172
HeparII-500	HeparII	500	70	123	−17 341.071
Win95pts-500	Win95pts	500	76	112	−4949.510
Ande-300	Ande	300	223	338	−30 166.843

4.5. Time complexity analysis

From the procedure of SIGA-BN, the inner loop steps are CHC selection, uniform crossover, elite learning strategy based on MB, v-structure-based adaptive mutation operator, and fitness evaluation. Assume that n is the number of nodes, m is the number of edges, N is population size, M is the number of training data, and Max_gen is the maximum number of generations. The time complexity of mutual independence is $O(nlog(n))$ and the time complexity of removing cycles is $O(m)$. The time complexity is $O(nlogn)$ for the selection and crossover operations. The fitness evaluation, which evaluates the BDeu score, is computed with the time complexity $O(Mn^2)$. Therefore, the total time complexity of SIGA-BN is $O(n^2 * N * Max\_gen)$.

5. Experiment and analysis

5.1. Datasets

Eight widely used BNs with different sizes are chosen from the Bayesian Network Repository for evaluation; these BNs are listed in Table 2, including a small BN with 8 nodes and 8 edges (ASIA) [48], a small BN with 11 nodes and 17 edges (SACHS) [49], a medium BN with 27 nodes and 52 edges (INSURANCE) [50], a medium BN with 37 nodes and 46 edges (ALARM) [51], a medium BN with 48 nodes and 84 edges (BARLEY) [52], a large BN with 70 nodes and 128 edges (HEPAR II) [53], a large BN with 76 nodes and 112 edges (WIN95PTS) containing 500 cases, and a very large network with 223 nodes and 338 edges (ANDE) [54] containing 300 cases.

5.2. Experimental settings

The effects of the proposed elite learning strategy and the v-structure-based adaptive mutation operator are first studied. Then, we evaluate the performance of SIGA-BN in searching the optimal network structure of a BN by comparing it with three GA-based BN structure learning algorithms (AESL-GA [25], EKGA-BN [26], and hybrid-SLA-GA [15]), five traditional algorithms, which include two score-based methods (the K2 algorithm [12] and HC algorithm [13]), two hybrid algorithms (MMHC algorithm [22] and SaiyanH algorithm [24]) and a constraint-based algorithm called the Inter-IAMB [28] algorithm. Finally, convergence rate and execution time comparisons are carried out.

For a fair comparison, the parameters of all the compared algorithms are configured as recommended in the references to obtain the best performance. Table 3 gives the parameter settings of the four GA-based algorithms. The maximum number of generations for the extremely large ANDE network is 500 due to its super-large network structure and parameters. The order of nodes for K2 is generated randomly. The BDeu score is used in the experiments as described in Section 3.1.

The HC and MMHC algorithms and the proposed algorithm are implemented in the BNT Structure Learning Package (BNT-SLP), and the Causal Explorer system is used. The MMHC algorithm is implemented in the Causal Explorer system [55]. The other algorithms are implemented with the BNT-SLP [56], developed on the Bayes Net Toolbox for MATLAB [57]. Each algorithm is executed 20 times independently for each dataset, and the average results are recorded.

Two servers are used in the experiment as follows.

1. The first server: an Intel(R) Core(TM) i5 CPU @ 2.3 GHz 8 GB of RAM,The operating system is macOS.
2. The second server:an Intel(R) Core(TM) i5 CPU @ 3.0 GHz 8 GB of RAM,The operating system is macOS.

The experiments on the datasets Asia, Sachs, Insurance, Alarm, and HeparII are running on the first server, and the experiments on the other datasets are running on the second server.

The performance of the algorithms is compared according to the score and structure differences between the final output network and the original network structure. The F1-score, sensitivity, and specificity [14] are the evaluation measures.

5.3. Ablation study for the proposed algorithm

Ablation study can be used to evaluate the performance of the components of the algorithms and is a helpful way to understand better the algorithms [58], which has also been used in EAs [59]. To evaluate the effectiveness of the proposed operators, ablation experiments are performed here on the ASIA, ALARM, and HEPAR II datasets. The standard GA with CHC selection is taken as the baseline algorithm. First, the elite learning strategy based on an MB is added after the selection operator. Then the single point mutation operator is replaced with the proposed v-structure-based adaptive mutation operator, which forms the proposed SIGA-BN. The experimental results with mean values and standard deviations are presented in Table 4.

As shown in Table 4, the average BDeu scores are the same for all three algorithms on the ASIA-500 dataset, which signifies that all versions can obtain the optimal results on the small dataset. From the results of the ablation study on the larger ALARM and HEPAR II datasets, it is clear that the F1 score and BDeu score performances are both improving, which verifies the effect of the proposed learning mechanism and adaptive mutation operator. From the perspective of standard deviations, SIGA-BN achieves the best results on all three datasets, which shows that SIGA-BN is much more stable due to the addition of the two proposed operators. Generally, both the elite learning strategy and v-structure-based adaptive mutation operator induce progress in terms of stability and structure accuracy.

5.4. Comparison between SIGA-BN and the other structure learning algorithms

The results yielded by nine algorithms on eight datasets after 20 independent runs are summarized in Table 5. The standard deviation is in parentheses, and the best results are shown in bold. In addition, the BDeu results and HD results were analyzed using the Wilcoxon rank sum test with a significance level of 0.05. The symbols '+', '−', '≈' indicate that the SIGA algorithm is significantly better than the comparison algorithms, worse than the comparison algorithms, or statistically similar to the comparison algorithm. The last two rows of the table summarize the number of symbols '+', '≈'. A higher F1 score indicates that the learned network is closer to the standard network. A smaller Hamming Distance means that the learned network is less different from the standard network.

For the small ASIA dataset, both the score-based and hybrid methods can produce BNs with higher scores than the constraint-based method and lower scores than those of the GA-based hybrid methods. The structures obtained by the constraint-based methods cannot fit the data appropriately since the data are not sufficient. ASIA's search space is small so the GA-based hybrid algorithms can reach the global optimum. Therefore, the GA-based hybrid algorithms are capable of finding

Table 3
Parameter settings for SIGA-BN and the GA-based algorithms.

Experiment	Population size	Maximum number of generation	Tournament size	CI test threshold	Elite eligibility threshold	Maximum parent node number	Elite learning rate	Elite size
SIGA-BN	100	100	N/A	0.01	N/A	4	0.5	10
AESL-GA	100	100	N/A	0.01	0.9	4	N/A	N/A
EKGA-BN	100	100	4	0.01	0.5	4	N/A	N/A
Hybrid-SLA-GA	100	100	N/A	0.01	N/A	4	N/A	N/A

Table 4
Ablation experiments for the proposed SIGA-BN (best results in bold).

Algorithms	Asia-500		Alarm-500		HeparII-500	
	F1 Score	BDeu score	F1 Score	BDeu score	F1 Score	BDeu score
Baseline	0.788 (0.057)	−1146.521 (0)	0.690 (0.068)	−6176.178 (38.88)	0.313 (0.020)	−16 421.237 (3.864)
Baseline + elite learning strategy	0.788 (0.062)	−1146.521 (0)	0.716 (0.087)	−6162.069 (51.715)	0.320 (0.015)	−16 420.363 (3.649)
Baseline + elite learning strategy + v-structure based adaptive mutation (SIGA-BN)	**0.813** (0.057)	−1146.521 (0)	**0.756** (0.060)	**−6128.878** (30.851)	**0.325** (0.014)	**−16 419.755** (3.284)

Table 5
Experimental results obtained by nine algorithms on eight datasets.

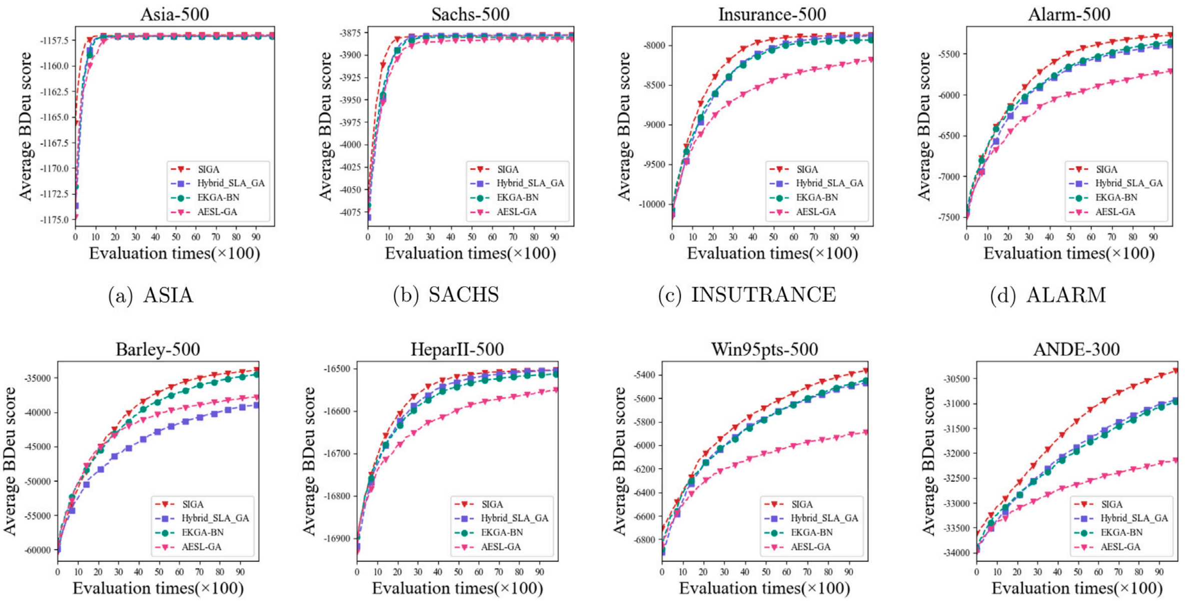
Method		Asia-500	Sachs-500	Insurance-500	Alarm-500	Barley-500	HeparII-500	Win95pts-500	ANDE-300
SIGA-BN	F1 Score	**0.881** (0.081)	**0.892** (0.041)	**0.758** (0.033)	**0.857** (0.045)	**0.392** (0.029)	**0.449** (0.028)	**0.565** (0.024)	**0.553** (0.011)
	Sensitivity	**0.837** (0.060)	**0.810** (0.063)	**0.637** (0.029)	**0.874** (0.046)	**0.292** (0.018)	**0.326** (0.027)	**0.547** (0.032)	**0.519** (0.011)
	Specificity	0.982 (0.029)	**0.999** (0.005)	**0.994** (0.005)	0.990 (0.004)	0.985 (0.004)	0.994 (0.001)	0.986 (0.002)	0.995 (0)
	HD	**2.138** (1.382)	**5.235** (1.540)	**22.175** (3.059)	**17.281** (4.902)	**76.45** (5.738)	**99.1** (3.77)	**96** (6.731)	285.467 (6.370)
	BDeu score	**−1157.1** (24.9)	**−3874.1** (40.4)	**−7877.2** (70.7)	**−5271.4** (149.7)	−33 430.9 (481.6)	**−16 473.7** (135.9)	−5287.4 (76.2)	−30 318.1 (160.8)
AESL-GA	F1 Score	0.877 (0.076)	0.872 (0.055)	0.691 (0.061)	0.719 (0.048)	0.330 (0.046)	0.410 (0.036)	0.481 (0.041)	0.446 (0.020)
	Sensitivity	0.833 (0.062)	0.788 (0.068)	0.574 (0.056)	0.742 (0.046)	0.234 (0.036)	0.292 (0.030)	0.437 (0.041)	0.392 (0.023)
	Specificity	0.982 (0.024)	0.996 (0.008)	0.987 (0.006)	0.979 (0.004)	0.987 (0.004)	0.993 (0.001)	0.986 (0.002)	0.995 (0.001)
	HD	2.225 (1.23)≈	5.416 (1.57)≈	27.725 (5.031)+	28.2 (4.681)+	79.65 (5.593)≈	103.9 (5.426)+	105.925 (7.897)+	329.525 (16.398)+
	BDeu score	**−1157.1** (24.006)≈	−3882.5 (38.4)≈	−7992.7 (161.1)+	−5451.4 (151.0)+	−35 473.8 (813.6)+	−16 520.5 (134.0)≈	−5641.4 (127.6)+	−31 529.2 (157.1)+
EKGA-BN	F1 Score	0.869 (0.086)	0.866 (0.059)	0.737 (0.035)	0.831 (0.044)	0.388 (0.046)	0.433 (0.034)	0.550 (0.034)	0.553 (0.014)
	Sensitivity	0.827 (0.065)	0.772 (0.089)	0.615 (0.036)	0.834 (0.047)	0.275 (0.034)	0.313 (0.029)	0.521 (0.034)	0.515 (0.012)
	Specificity	0.979 (0.031)	0.998 (0.006)	0.992 (0.004)	0.989 (0.003)	0.990 (0.003)	0.994 (0.001)	**0.986** (0.002)	0.995 (0)
	HD	2.235 (1.46)≈	5.8 (1.639)≈	23.725 (2.972)≈	19 (3.553)≈	73.3 (5.543)≈	101.3 (5.053)≈	86.167 (6.984)≈	**282.590** (5.38)≈
	BDeu score	**−1157.1** (24.0)≈	−3879.6 (38.5)≈	−7905.8 (91.5)≈	−5272.6 (145.3)≈	−33 634.6 (586.8)≈	−16 508.1 (134.6)≈	−5292.1 (64.6)+	**−30 130.6** (92.7)+
Hybrid-SLA-GA	F1 Score	0.872 (0.087)	0.869 (0.053)	0.745 (0.040)	0.738 (0.060)	0.267 (0.029)	0.442 (0.036)	0.498 (0.025)	0.490 (0.017)
	Sensitivity	0.827 (0.065)	0.777 (0.077)	0.625 (0.037)	0.789 (0.049)	0.217 (0.023)	0.319 (0.032)	0.497 (0.022)	0.466(0.018)
	Specificity	0.980 (0.031)	0.998 (0.006)	0.992 (0.005)	0.978 (0.006)	0.970 (0.006)	0.994 (0.001)	0.982 (0.002)	0.994 (0)
	HD	2.3 (1.45)+	6.075 (1.477)≈	23.45 (3.626)+	28.35 (6.067)+	99.775 (6.767)+	99.775 (4.841)≈	112.575 (7.066)+	328.42512.227)+
	BDeu score	**−1157.1** (24.0)≈	3878.4 (38.4)≈	−7887.5 (76.2)≈	−5380.6 (161.1)≈	−38 856.6 (1052.4)+	−16 504.4 (134.3)≈	−5464.6 (108.2)+	−30 918.1(161.4)+
K2	F1 Score	0.623 (0.104)	0.777 (0.114)	0.538 (0.070)	0.607 (0.063)	0.312 (0.025)	0.325 (0.041)	0.363 (0.034)	0.397 (0.015)
	Sensitivity	0.690 (0.118)	0.708 (0.120)	0.460 (0.057)	0.761 (0.063)	0.215 (0.020)	0.249 (0.034)	0.545 (0.049)	0.459(0.015)
	Specificity	0.873 (0.044)	0.970 (0.023)	0.965 (0.011)	0.958 (0.008)	0.988 (0.002)	0.986 (0.003)	0.950 (0.005)	0.990 (0.001)
	HD	7 (1.62)+	8.45 (3.057)+	42.225 (6.082)+	45.675 (7.126)+	79.925 (2.399)+	126.75 (7.965)+	200.3 (14.824)+	456.725 (15.629)+
	BDeu score	−1165.6 (23.4)≈	−3969.8 (72.7)+	−8483.4 (176.0)+	−5552.5 (146.4)+	34 530.196 (503.9)≈	−16 585.134 (142.282)≈	**−5221.9** (63.0)≈	−30 374.6(168.2)≈
HC	F1 Score	0.802 (0.130)	0.881 (0.049)	0.665 (0.058)	0.754 (0.027)	0.382 (0.040)	0.392 (0.032)	Running out of time	Running out of time
	Sensitivity	0.814 (0.117)	0.797 (0.073)	0.566 (0.056)	0.873 (0.024)	0.282 (0.033)	0.325 (0.032)		
	Specificity	0.948 (0.043)	0.997 (0.007)	0.981 (0.006)	0.976 (0.004)	0.987 (0.003)	0.984 (0.002)		
	HD	3.825 (2.215)+	5.475 (1.453)≈	31.625 (5.061)+	20.636 (3.503)+	77.025 (4.44)≈	123.6 (5.447)+		
	BDeu score	−1158.5 (24.9)≈	−3890.8 (39.145)≈	−7996.2 (103.3)+	−5117.5 (132.8)≈	**−32 941.2** (335.9)≈	−16 485.0 (136.1)≈		
SaiyanH	F1 Score	0.692 (0.114)	0.548 (0.067)	0.479 (0.082)	0.707 (0.051)	Running out of time	0.275 (0.058)	0.319 (0.029)	Running out of time
	Sensitivity	0.688 (0.122)	0.485 (0.061)	0.409 (0.073)	0.664 (0.051)		0.247 (0.028)	0.367 (0.031)	
	Specificity	0.954 (0.020)	0.980 (0)	0.977 (0.007)	**0.997** (0.002)		0.974 (0.018)	0.970 (0.003)	
	HD	4.85 (1.711)+	13.6 (2.035)+	46.25 (7.354)+	25.3 (4.394)+		169.05 (50.087)+	176 (11.362)+	
	BDeu score	−1158.838 (24.2)≈	−3912.8 (38.6)+	−8423.9 (189.5)+	−5326.7 (138.2)≈		**−17 095.5** (705.8)+	−5486.09 (139.1)+	
MMHC	F1 Score	0.687 (0.134)	0.546 (0.113)	0.568 (0.076)	0.664 (0.098)	0.384 (0.084)	0.299 (0.043)	0.366 (0.055)	0.495 (0.034)
	Sensitivity	0.638 (0.118)	0.524 (0.111)	0.475 (0.066)	0.663 (0.094)	0.290 (0.064)	0.208 (0.031)	0.339 (0.053)	0.430 (0.029)
	Specificity	0.973 (0.030)	**0.999** (0.004)	0.993 (0.004)	0.992 (0.003)	**0.992** (0.001)	**0.995** (0.001)	**0.987** (0.002)	**0.997** (0)
	HD	4.7 (2.193)+	14.8 (3.682)+	37.5 (6.454)+	30.95 (9.211)+	78.05 (10.278)+	119.95 (7.166)+	131.5 (11.948)≈	297.05(20.946)+
Inter-IAMB	F1 Score	0.51 (0.114)	0.518 (0.070)	0.325 (0.032)	0.627 (0.037)	0.179 (0.024)	0.190 (0.027)	0.366 (0.025)	0.336 (0.017)
	Sensitivity	0.355 (0.104)	0.800 (0.078)	0.335 (0.040)	0.588 (0.042)	0.157 (0.022)	0.115 (0.017)	0.233 (0.019)	0.261 (0.013)
	Specificity	**0.998** (0.008)	0.653 (0.061)	0.897 (0.012)	0.983 (0.003)	0.957 (0.004)	0.995 (0.003)	0.988 (0.001)	0.996 (0)
	HD	5.5 (0.894)+	23.5 (3.761)+	70.05 (3.879)+	34.275 (2.861)+	118.925 (4.539)≈	120.825 (5.994)+	92.075 (2.899)+	347.45 (11.268)≈
+/−/≈	HD	6/0/2	4/0/4	7/0/1	7/0/1	3/0/4	6/0/2	5/0/2	4/0/2
	BDeu	0/0/6	2/0/4	4/0/2	2/0/4	2/0/3	1/0/5	4/0/1	3/0/1

the global optima in small networks, while traditional score-based algorithms may easily fall into local optima.

The proposed SIGA-BN achieves the best scores on the other seven datasets compared with the other algorithms. The score of HC algorithm in BARLEY-500 gets the best scores, but the structural accuracy is worse than SIGA-BN. As the search space increases, the HC algorithm runs out of time and cannot obtain a solution. SIGA-BN can obtain a similar score on HEPARII-500 and a better BN with better scores on WIN95PTS-500 and ANDE-300.

Regarding the BDeu scores, compared with the constraint-based algorithms, the proposed SIGA-BN can obtain a structure that is more suitable for the data when the sample size is small. When the search space is large, SIGA-BN can obtain a better solution in a limited timeframe with a higher score than the constraint-based algorithms. A BN structure with a higher score and the best HD can be obtained by SIGA-BN relative to the other GA-based hybrid algorithms.

From the perspective of structure accuracy, the proposed SIGA-BN can find a BN with the highest accuracy. Due to the small sample size, it is impossible for constraint-based methods to obtain a BN network with high accuracy in cases with low CI test accuracy. Among the score-based algorithms, the HC algorithm cannot obtain a feasible solution within the limited timeframe when the search space becomes huge. Although the score-based method can find BNs with high scores for small and medium-sized networks, its structural accuracy is low, so it cannot accurately reflect causality and cannot fit the data appropriately. The proposed SIGA-BN is the best-performing algorithm for maximizing the F1 score and sensitivity. In contrast, the MMHC algorithm performs well concerning specificity on large networks like BARLEY-500, HEPARII-500, WIN95PTS-500, and ANDE-300 datasets.



(a) ASIA (b) SACHS (c) INSUTRANCE (d) ALARM

(e) BARLEY (f) HEPARII (g) WIN95PTS (h) ANDE

Fig. 6. Convergence rate of different algorithms in benchmark networks.

But the SaiyanH algorithm will run out of time when dealing the datasets with many parameters like BARLEY-500 and ANDE-300.

Due to the limited amount of data and the high correlation among the constraint-based algorithms regarding the accuracy obtained on the CI test, the F1 and BDeu scores of the Inter-IAMB algorithm are poor, indicating that constraint-based algorithms may perform poorly when the training data are insufficient. The K2 and HC algorithms are both efficient single-solution search methods. K2 optimizes the BDeu score effectively, but the final solution does not match the original BN well and therefore yields a low F1 score. The HC algorithm can identify BN structures with lower BDeu scores than those of K2 on ASIA-500, ALARM-500, and BARLEY-500, and get better HD score but runs out of time when working with larger datasets.

The F1 score of MMHC is best among those of the K2, HC, and Inter-IAMB algorithms, which means that the BN structure identified by MMHC is closest to the representation of reality, and thus, the accuracy is highest. However, its F1 scores are worse than the GA-based algorithms on large datasets. From the statistical results on HD and BDeu, it is clear that the proposed SIGA-BN has achieved the best performance in almost all the cases.

5.5. Comparison on the convergence speed

The convergence speeds of the four GA-based algorithms on eight networks are plotted in Fig. 6. As the number of variables increases, the convergence speeds of the AESL-GA algorithm decrease. For BARLEY-500, the SaiyanH algorithm performs worse than other datasets with fewer parameters. For ANDE-300, the structural information of the BN is used to guide the search process; thus, the exploration ability of SIGA-BN is enhanced, and the resulting score is significantly higher than that of the other GA-based methods. The convergence speed of SIGA-BN is faster than that of the other GA-based algorithms on all datasets, and the proposed algorithm is most likely to obtain the highest BDeu score. The GA-based algorithms need more iterations to obtain better solutions, which leads to excessive execution times, especially when the given network is extremely large. SIGA-BN can be used to find a BN with a more accurate structure more efficiently.

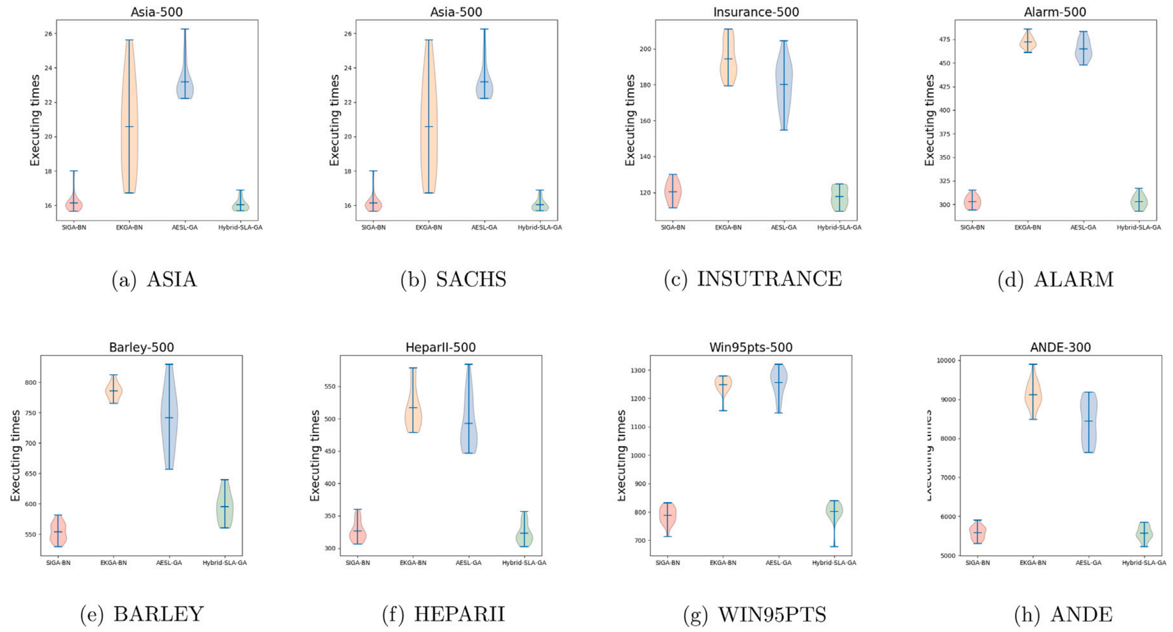
5.6. Comparison on the computational time

The computational times of the four GA-based algorithms on eight networks are plotted in Fig. 7. Due to the large gap in the structure accuracy, only the GA-based hybrid algorithms are chosen, and the other algorithms are not in this comparison. As shown in Fig. 7, the proposed SIGA-BN requires the least computational time and achieves the lowest standard deviations on the eight datasets. SIGA-BN uses the structural information of a BN to guide the search process, which is different from the commonly used GA-based methods, and then the computational time is reduced. At the same time, SIGA-BN converges faster and can find a solution with fewer iterations, which significantly decreases the time required for score calculation and searching. The algorithm's shorter computational time indicates that SIGA-BN can complete the search process in a shorter time and find a BN with higher accuracy.

Fig. 8 shows the average computational time of SIGA-BN compared to those of the other GA-based hybrid algorithms on the ALARM network and HEPARII network with five different sample sizes. It can be seen from the experimental results that the computational times of SIGA-BN and hybrid-SLA-GA generally increase with increasing the sample size, and the running time is significantly less than the other two algorithms. The performance of EKGA-BN and AESL-GA algorithms on different datasets has little difference. Notably, the proposed SIGA-BN takes less time to learn proper BNs in all cases. Therefore, we believe that the proposed SIGA-BN algorithm is more efficient than the other GA-based algorithms in terms of computational time.

5.7. Discussion

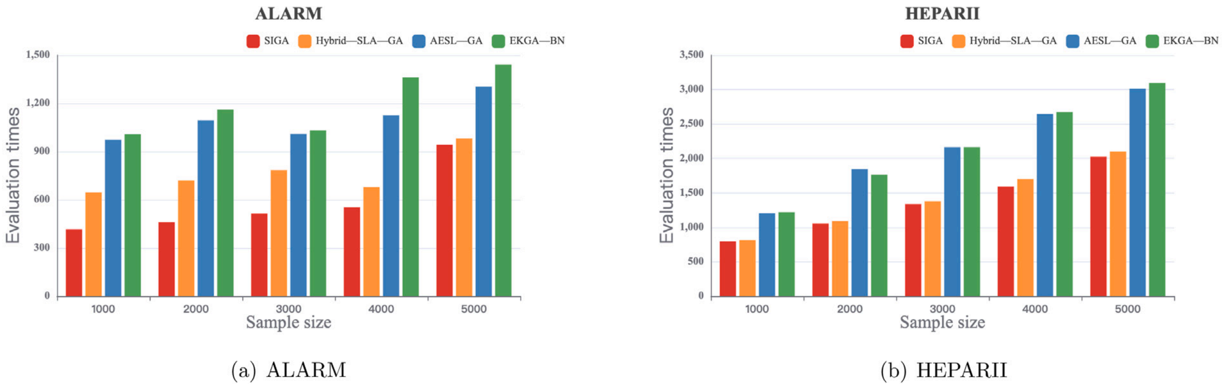
Advantages: The GA-based algorithms can find accurate BN structures for datasets with few variables, while traditional methods have difficulty obtaining reasonable solutions. With the growth in the number of nodes, the score-based methods can find BNs with higher scores than those of the constraint-based methods, but the score-based methods take more time than the GA-based methods and may even fail to obtain a feasible solution with a limited time-frame for vast networks. Compared with the other GA-based methods, the proposed SIGA-BN can obtain solutions with better scores in less time. The faster convergence speed and better exploration ability of the proposed method make the



(a) ASIA (b) SACHS (c) INSUTRANCE (d) ALARM

(e) BARLEY (f) HEPARII (g) WIN95PTS (h) ANDE

Fig. 7. Computational time of different algorithms on benchmark networks.



(a) ALARM (b) HEPARII

Fig. 8. Computational times of the GA-based algorithms on ALARM and HEPARII with five sample sizes.

search process take less time for a large network and yield a more accurate BN structure. The proposed elite learning strategy introduces the idea of an MB to use the structural information of elite individuals better. The v-structure-based adaptive mutation operator reduces the redundancy of the search process through the concept of a v-structure and balances exploration and exploitation according to the adaptive preference-changing mechanism. Compared with other widely used BN structure learning algorithms, SIGA-BN can find better BNs more effectively.

Disadvantages: In order to get a more accurate BN structure, one possible way is to increase the number of iterations which leads to more fitness evaluations. Another possible way is to increase the number of samples. Since the computational burden of the proposed SIGA-BN heavily depends on the number of fitness evaluations. The running time of both ways on the proposed algorithm is therefore increased. At the same time, with the increasing number of nodes and arcs, the computational time of SIGA-BN is increased quickly. The parallel realization for BNSL with GA based on CPU or GPU can help to address the time-consuming problem [60,61]. The proposed elite learning strategy base on MB has shown its effectiveness. However, compared with the other GA-based BNSL algorithms, there are two more parameters for the elite learning strategy in SIGA-BN, which is inconvenient to some extent.

Future works: For the elite learning strategy in GA, the individuals with better fitness values are usually regarded as elite. However, for BNSL, we think the structure information in BNs can be further incorporated with fitness value to determine an elite individual, which helps to evaluate the individual more comprehensively. In addition, since BN is an essential tool for causal analysis, we plan to use the obtained BNs by SIGA-BN in real-world applications, such as food safety [62,63].

6. Conclusion

This paper introduces SIGA-BN for efficiently solving the BN structure learning problem. We first introduce the concept of MBs in BNs to represent the characteristics of individuals. Then, an elite learning strategy is designed to better learn the structural information of the elite individuals, which helps to accelerate the convergence speed with high accuracy. The concept of v-structures in BNs is utilized to design an adaptive mutation operator to reduce the search process's redundancy and balance the model's exploration and exploitation capabilities. Experimental results show that the performance of GA-based algorithms is generally better than those of single solution, score-based search, and constraint-based algorithms. Compared with the other GA-based BN structure learning algorithms, the proposed SIGA-BN is capable of constructing near-optimal networks with higher BDeu scores and faster convergence speed. SIGA-BN also exhibits robust performance and provides significant results, especially for datasets with many variables.

CRediT authorship contribution statement

Wei Fang: Supervision. **Weijian Zhang:** Writing – original draft, Methodology. **Li Ma:** Writing. **Yunlin Wu:** Writing. **Kefei Yan:** Writing. **Hengyang Lu:** Resource. **Jun Sun:** Resource. **Xiaojun Wu:** Funding acquisition. **Bo Yuan:** Language editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported in part by the National Natural Science foundation of China under Grant 62073155, 62002137, 62106088, and 62206113, in part by ‘‘Blue Project’’ in Jiangsu Universities, China, in part by Innovative Research Foundation of Ship General Performance, China under Grant 22422213, in part by Guangdong Provincial Key Laboratory, China under Grant 2020B121201001.

References

[1] S. Gheisari, M.R. Meybodi, Bnc-pso: structure learning of bayesian networks by particle swarm optimization, Inform. Sci. 348 (2016) 272–289.

[2] J. Xuan, J. Lu, G. Zhang, R.Y. Da Xu, X. Luo, A Bayesian nonparametric model for multi-label learning, Mach. Learn. 106 (11) (2017) 1787–1815.

[3] F.L. Seixas, B. Zadrozny, J. Laks, A. Conci, D.C.M. Saade, A Bayesian network decision model for supporting the diagnosis of dementia, Alzheimer’s disease and mild cognitive impairment, Comput. Biol. Med. 51 (2014) 140–158.

[4] S. Nikolopoulos, G.T. Papadopoulos, I. Kompatsiaris, I. Patras, Evidence-driven image interpretation by combining implicit and explicit knowledge in a bayesian network, IEEE Trans. Syst. Man Cybern. B 41 (5) (2011) 1366–1381.

[5] P. LarrañAga, H. Karshenas, C. Bielza, R. Santana, A review on evolutionary algorithms in Bayesian network learning and inference tasks, Inform. Sci. 233 (2013) 109–125.

[6] K. Pearson, X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling, Lond. Edinb. Dublin Philos. Mag. J. Sci. 50 (302) (1900) 157–175.

[7] P. Spirtes, C.N. Glymour, R. Scheines, D. Heckerman, Causation, Prediction, and Search, MIT Press, 2000.

[8] P. Spirtes, C. Glymour, An algorithm for fast recovery of sparse causal graphs, Soc. Sci. Comput. Rev. 9 (1) (1991) 62–72.

[9] J.-P. Pellet, A. Elisseeff, Using Markov blankets for causal structure learning, J. Mach. Learn. Res. 9 (Jul) (2008) 1295–1342.

[10] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Mach. Learn. 29 (2–3) (1997) 131–163.

[11] C. Chow, C. Liu, Approximating discrete probability distributions with dependence trees, IEEE Trans. Inform. Theory 14 (3) (1968) 462–467.

[12] G.F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Mach. Learn. 9 (4) (1992) 309–347.

[13] W.L. Buntine, Operations for learning with graphical models, J. Artificial Intelligence Res. 2 (1994) 159–225.

[14] F. Vafaee, Learning the structure of large-scale Bayesian networks using genetic algorithm, in: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, ACM, 2014, pp. 855–862.

[15] S. Jose, S. Liu, S. Louis, S. Dascalu, Towards a hybrid approach for evolving Bayesian networks using genetic algorithms, in: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence, ICTAI, IEEE, 2019, pp. 705–712.

[16] S. Jose, S.J. Louis, S.M. Dascalu, S. Liu, Bayesian network structure learning using case-injected genetic algorithms, in: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence, ICTAI, IEEE, 2020, pp. 572–579.

[17] J. Wang, S. Liu, Novel binary encoding water cycle algorithm for solving Bayesian network structures learning problem, Knowl.-Based Syst. 150 (2018) 95–110.

[18] K. Liu, Y. Cui, J. Ren, P. Li, An improved particle swarm optimization algorithm for Bayesian network structure learning via local information constraint, IEEE Access 9 (2021) 40963–40971.

[19] L.M. De Campos, J.M. Fernandez-Luna, J.A. Gámez, J.M. Puerta, Ant colony optimization for learning Bayesian networks, Internat. J. Approx. Reason. 31 (3) (2002) 291–311.

[20] P. Larrañaga, M. Poza, Y. Yurramendi, R.H. Murga, C.M.H. Kuijpers, Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters, IEEE Trans. Pattern Anal. Mach. Intell. 18 (9) (1996) 912–926.

[21] R. Kabli, F. Herrmann, J. McCall, A chain-model genetic algorithm for Bayesian network structure learning, in: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, ACM, 2007, pp. 1264–1271.

[22] I. Tsamardinos, L.E. Brown, C.F. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, Mach. Learn. 65 (1) (2006) 31–78.

[23] N. Friedman, I. Nachman, D. Peér, 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., 1999, pp. 206–215.

[24] A.C. Constantinou, Learning Bayesian networks that enable full propagation of evidence, IEEE Access 8 (2020) 124845–124856.

[25] C. Contaldi, F. Vafaee, P.C. Nelson, Bayesian network hybrid learning using an elite-guided genetic algorithm, Artif. Intell. Rev. 52 (1) (2019) 245–272.

[26] W. Zhang, W. Fang, J. Sun, Q. Chen, Learning Bayesian networks structures with an effective knowledge-driven GA, in: 2020 IEEE Congress on Evolutionary Computation, CEC, IEEE, 2020, pp. 1–8.

[27] J. Cheng, D. Bell, W. Liu, Learning Bayesian networks from data: An efficient approach based on information theory, 1998, On World Wide Web at http://www.cs.ualberta.ca/∼jcheng/bnpc.htm.

[28] I. Tsamardinos, C.F. Aliferis, A.R. Statnikov, E. Statnikov, Algorithms for large scale Markov blanket discovery, in: FLAIRS Conference, Vol. 2, St. Augustine, FL, 2003, pp. 376–380.

[29] X. Qi, X. Fan, Y. Gao, Y. Liu, Learning Bayesian network structures using weakest mutual-information-first strategy, Internat. J. Approx. Reason. 114 (2019) 84–98.

[30] S. Behjati, H. Beigy, Improved K2 algorithm for Bayesian network structure learning, Eng. Appl. Artif. Intell. 91 (2020) 103617.

[31] R. Xu, S. Liu, Q. Zhang, Z. Yang, J. Liu, PEWOBS: An efficient Bayesian network learning approach based on permutation and extensible ordering-based search, Future Gener. Comput. Syst. 128 (2022) 505–520.

[32] Q. Ye, A.A. Amini, Q. Zhou, Optimizing regularized Cholesky score for order-based learning of Bayesian networks, IEEE Trans. Pattern Anal. Mach. Intell. 43 (10) (2021) 3555–3572.

[33] B. Sun, Y. Zhou, J. Wang, W. Zhang, A new PC-PSO algorithm for Bayesian network structure learning with structure priors, Expert Syst. Appl. 184 (2021) 115237.

[34] X. Wang, H. Ren, X. Guo, A novel discrete firefly algorithm for Bayesian network structure learning, Knowl.-Based Syst. 242 (2022) 108426.

[35] M. Scanagatta, A. Salmerón, F. Stella, A survey on Bayesian network structure learning from data, Prog. Artif. Intell. 8 (4) (2019) 425–439.

[36] P. Larrañaga, H. Karshenas, C. Bielza, R. Santana, A review on evolutionary algorithms in Bayesian network learning and inference tasks, Inform. Sci. 233 (2013) 109–125.

[37] I. Tsamardinos, C.F. Aliferis, A. Statnikov, Time and sample efficient discovery of Markov blankets and direct causal relations, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 673–678.

[38] G. Caravagna, D. Ramazzotti, Learning the structure of Bayesian networks via the bootstrap, Neurocomputing 448 (2021) 48–59.

[39] J. Dai, J. Ren, W. Du, Decomposition-based Bayesian network structure learning algorithm using local topology information, Knowl.-Based Syst..

[40] C. Song, Y. Zhang, Z. Xu, An improved structure learning algorithm of Bayesian network based on the hesitant fuzzy information flow, Appl. Soft Comput. 82 (2019) 105549.

[41] H. Akaike, A new look at the statistical model identification, in: Selected Papers of Hirotugu Akaike, Springer, 1974, pp. 215–222.

[42] Y.M. Shtar’kov, Universal sequential coding of single messages, Probl. Pereda. Inf. 23 (3) (1987) 3–17.

[43] J. Suzuki, A construction of Bayesian networks from databases based on an MDL principle, in: Uncertainty in Artificial Intelligence, Elsevier, 1993, pp. 266–273.

[44] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.

[45] L.M. Aplin, B.C. Sheldon, J. Morand-Ferron, Milk bottles revisited: social learning and individual variation in the blue tit, Cyanistes caeruleus, Anim. Behav. 85 (6) (2013) 1225–1232.

[46] R. Cheng, Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Inform. Sci. 291 (2015) 43–60.

[47] Q. Yang, W.-N. Chen, T. Gu, H. Zhang, J.D. Deng, Y. Li, J. Zhang, Segment-based predominant learning swarm optimizer for large-scale optimization, IEEE Trans. Cybern. 47 (9) (2017) 2896–2910, http://dx.doi.org/10.1109/TCYB.2016.2616170.

[48] S.L. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, J. R. Stat. Soc. Ser. B Stat. Methodol. 50 (2) (1988) 157–194.

[49] K. Sachs, O. Perez, D. Pe'er, D.A. Lauffenburger, G.P. Nolan, Causal protein-signaling networks derived from multiparameter single-cell data, Science 308 (5721) (2005) 523–529.

[50] J. Binder, D. Koller, S.J. Russell, K. Kanazawa, Adaptive probabilistic networks with hidden variables, Mach. Learn. 29 (2004) 213–244.

[51] I.A. Beinlich, H.J. Suermondt, R.M. Chavez, G.F. Cooper, The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks, in: AIME 89, Springer, 1989, pp. 247–256.

[52] K. Kristensen, I.A. Rasmussen, The use of a Bayesian network in the design of a decision support system for growing malting barley without use of pesticides, Comput. Electron. Agric. 33 (3) (2002) 197–217.

[53] A. Onisko, Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders (Ph. D. dissertation), Inst. Biocybern. Biomed. Eng., Polish Academy Sci., Warsaw, Poland, 2003.

[54] C. Conati, A.S. Gertner, K. VanLehn, M.J. Druzdzel, On-line student modeling for coached problem solving using Bayesian networks, in: User Modeling, Springer, 1997, pp. 231–242.

[55] C.F. Aliferis, I. Tsamardinos, A.R. Statnikov, L.E. Brown, Causal explorer: A causal probabilistic network learning toolkit for biomedical discovery, in: METMBS, Vol. 3, Citeseer, 2003, pp. 371–376.

[56] P. Leray, O. Francois, BNT Structure Learning Package: Documentation and Experiments, Tech. Rep, Laboratoire PSI, Universitè et INSA de Rouen, 2004.

[57] K.P. Murphy, The bayes net toolbox for matlab, Comput. Sci. Stat. 33 (2001) 2001.

[58] C. Fawcett, H.H. Hoos, Analysing differences between algorithm configurations through ablation, J. Heuristics 22 (4) (2016) 431–458.

[59] Z. Lu, I. Whalen, V. Boddeti, Y. Dhebar, K. Deb, E. Goodman, W. Banzhaf, Nsga-net: neural architecture search using multi-objective genetic algorithm, in: Proceedings of the Genetic and Evolutionary Computation Conference, 2019, pp. 419–427.

[60] Y. Wang, W. Qian, S. Zhang, X. Liang, B. Yuan, A learning algorithm for Bayesian networks and its efficient implementation on GPUs, IEEE Trans. Parallel Distrib. Syst. 27 (1) (2015) 17–30.

[61] S. Lee, S.B. Kim, Parallel simulated annealing with a greedy algorithm for Bayesian network structure learning, IEEE Trans. Knowl. Data Eng. 32 (6) (2019) 1157–1166.

[62] F. Rezazade, J. Summers, D.O. Lai Teik, A holistic approach to food fraud vulnerability assessment, Food Control 131 (2022) 108440, http://dx.doi.org/10.1016/j.foodcont.2021.108440, URL: https://www.sciencedirect.com/science/article/pii/S0956713521005788.

[63] H.J. Marvin, Y. Bouzembrak, E.M. Janssen, H.v. van der Fels-Klerx, E.D. van Asselt, G.A. Kleter, A holistic approach to food safety risks: Food fraud as an example, Food Res. Int. 89 (2016) 463–470.