

Research Article

Dynamic Programming Structure Learning Algorithm of Bayesian Network Integrating MWST and Improved MMPC

Ruo-Hai Di,^{1,2} Ye Li,² Ting-Peng Li,¹ Lian-Dong Wang,¹ and Peng Wang^{1,2} 

¹State Key Laboratory of Complex Electromagnetic Environment Effects on Electronic and Information System (CEMEE),
Luoyang 471003, China

²School of Electronics and Information Engineering, Xi'an Technological University, Xi'an 710021, China

Correspondence should be addressed to Peng Wang; wp_xatu@163.com

Received 25 July 2021; Revised 9 November 2021; Accepted 1 December 2021; Published 31 December 2021

Academic Editor: Tudor Barbu

Copyright © 2021 Ruo-Hai Di et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Dynamic programming is difficult to apply to large-scale Bayesian network structure learning. In view of this, this article proposes a BN structure learning algorithm based on dynamic programming, which integrates improved MMPC (maximum-minimum parents and children) and MWST (maximum weight spanning tree). First, we use the maximum weight spanning tree to obtain the maximum number of parent nodes of the network node. Second, the MMPC algorithm is improved by the symmetric relationship to reduce false-positive nodes and obtain the set of candidate parent-child nodes. Finally, with the maximum number of parent nodes and the set of candidate parent nodes as constraints, we prune the parent graph of dynamic programming to reduce the number of scoring calculations and the complexity of the algorithm. Experiments have proved that when an appropriate significance level α is selected, the MMPCDP algorithm can greatly reduce the number of scoring calculations and running time while ensuring its accuracy.

1. Introduction

In recent years, big data, machine learning, and deep learning have become hot spots of common concern in academia and industry, such as computer science, medicine, statistics, economy, and social sciences [1–5]. However, artificial intelligence scholars proposed at the annual conference of artificial intelligence that future artificial intelligence research should be oriented to uncertain environments and oriented to human-like mechanisms, and focus on dealing with complex problems and limited data learning problems. The “black box” design of the deep neural network model makes it difficult to explain its internal operating mechanism, and training the network requires a large number of labeled samples. It is not the only way to achieve future intelligence [6]. The methods that can deal with uncertainty and combine domain knowledge to model complex problems, such as fuzzy neural network [7, 8], Bayesian network [9], and DS evidence theory [10, 11], have attracted people's attention again.

Among them, the Bayesian network has many advantages that other modeling methods do not have. With its rigorous mathematical foundation, graphical topology that is easy to understand intuitively, and the natural expression of real problems, it has become the powerful tool for uncertain information processing and posterior probabilistic reasoning, which has been widely used in genetic analysis [12], medical diagnosis [13], reliability analysis [14], and threat assessment [15].

2. Related Work

When BN is used to deal with real problems, the structure and parameters of the model need to be given as inputs. The task of learning Bayesian network parameters is performed with the model structure obtained in advance, so it is crucial to learn the correct BN structure from the data. However, BN structure learning (BNSL) has been proved to be a nondeterministic polynomial hard problem. For a BN network with n nodes, the number of possible structures is

$O(n!2^{(n/2)})$ [16]. In DAG search space, BNSL can be divided into approximate learning algorithm and accurate learning algorithm according to learning accuracy. The former uses heuristic algorithms (ant colony algorithm [17], genetic algorithm [18], particle swarm optimization [19], etc.) for structure search, which is easy to fall into local optimum, and the structure finally obtained by learning is an approximate structure. The latter can guarantee the optimal network through the following methods: integer linear programming (ILP) [20], A^* algorithm [21], and dynamic programming (DP) [22]. The ILP algorithm decomposes BN structure learning into the problem of linear programming and then uses the SCIP solver to solve it. The core problem is how to choose cutting planes. A^* algorithm expresses BNSL as the shortest path programming problem [23–25], which uses the h value of nodes as the basis of path expansion to reduce the search space and the memory consumption of the algorithm. The DP algorithm decomposes BNSL into a problem of finding the best subnetworks, which recursively traverse the order space and parent space. It has been proved to be applicable to the network with no more than 26 nodes [26].

Dynamic programming algorithm is effective in small- and medium-scale BNSL, but it still has considerable complexity. To solve this problem, scholars at home and abroad have performed relevant research. Silander [26] proposed the SMDP algorithm, which reduces its running time by constructing P-caches storage structure and expands the computable network scale, but local scores are still required to be calculated for $n2^{n-1}$ times. Malone [27] proposed the memory-efficient dynamic programming algorithm (MEDP), which uses the properties of scoring functions to constrain the maximum number of parent nodes of a node and then adopts layer-by-layer storage calculation for dynamic programming, thus improving its efficiency by reducing a small number of scoring times. Singh [28] proposed an algorithm called dynamic programming based on depth-first search (DFS DP), which performs pruning operations to reduce storage space easily, but it has low computational efficiency. Ye [29] proposed a dynamic programming structure learning method integrating prior information. This method introduces edge constraints and path constraints to prune the order graph, thus reducing its time consumption. But it has low fault tolerance because its accuracy depends on the correctness of prior information. Tan et al. [30] proposed an algorithm of dynamic programming constrained with Markov blanket (DPCMB), which uses the Markov blanket calculated by the IAMB algorithm to constrain the scoring calculation process to reduce the number of scoring calculations. However, the significance level ϵ will affect its accuracy and efficient. Combining the hierarchical idea and dynamic programming, Yang [31] proposed a hierarchical optimal BN structure learning algorithm, which uses conditional independence to layer nodes and considers that the parent node of each node only comes from its node layer and the layer with higher priority. Behjati and Beigy [32] proposed to use the strongly connected graph to obtain the node block order, then use the DP algorithm to obtain the complete node

order, and finally obtain the BN structure through the K2 algorithm. Xu et al. [33] proposed a Bayesian network structure learning algorithm based on full permutation and extensible ordering-based search, which uses the local learning and pruning method to process the dataset with small and large number of variables, respectively, and obtains the candidate parent sets of each variable efficiently. Wang et al. [34] proposed a novel approach to improve the capability of a local search by determining the search direction, which can accelerate the convergence of the local search and acquire the higher quality structure. Gao et al. [35] proposed an improved K-means block learning Bayesian network structure algorithm, which uses the K-means algorithm fused with mutual information to block the network. Then, it adopts DP to learn the structure of each block and finally uses combination rules to obtain the best network. Yuan et al. [36] proposed to apply an improved A^* algorithm in the dynamic programming space to reduce the complexity of space and time; Liu [37] proposed a new optimal path selection based on a hybrid improved A^* algorithm and reinforcement learning method and obtained stable and efficient application effects in the optimal path selection of intelligent driving vehicles; Tan et al. [38] proposed a bidirectional heuristic search algorithm (BiHS) based on one-way heuristic search, and the results showed the BiHS algorithm is more efficient than the one-way heuristic search algorithm. Wang et al. [39] proposed the ancestor constrained ordered graph (ACOG), which improves the efficiency of precise search by introducing ancestor constraints.

To sum up, the existing literature has improved the DP algorithm by adding constraints or adopting spatial search strategies to reduce the time consumption to varying degrees. However, how to keep a balance between time consumption and algorithm accuracy is still a problem to be studied. In this article, the dynamic programming structure learning algorithm of Bayesian network integrating MWST and improved MMPC is proposed. By analyzing the connection relationship of nodes in maximum weight spanning tree, we take the maximum number of node connections in the tree structure as the maximum parent number t of nodes and prune the parent graph of nodes in the DP algorithm horizontally. Consequently, the upper limit of the parent number of nodes is reduced from $n-1$ to t . Then, we take the candidate parent-child node set **CPC** obtained by the MMPC algorithm as the candidate parent set **PS** of the corresponding node and prune the parent graph of nodes in the DP algorithm longitudinally. Meanwhile, to avoid the false-positive nodes in **CPC** (candidate parents and children), the MMPC algorithm is improved by the symmetry principle. With the above two operations, the time consumption and complexity of the algorithm can be lowered because the local scoring calculation times are reduced.

3. Principle of Dynamic Programming

From a mathematical point of view, the BN is a graph model representing the probability relationship among different random variables $V = \{X_1, X_2, \dots, X_n\}$, which consists of

structure \mathbf{G} and distribution parameter θ . Structure $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is a directed acyclic graph (DAG). Random variable \mathbf{V} is regarded as a node set. Directed edge set $E = \{e_{ij} | X_i \longrightarrow X_j\}$ shows the dependency relationship between X_i and X_j ; X_i is the parent set of X_j , that is, $X_i \in \text{pa}(X_j)$; X_j is the subset of X_i , that is, $X_j \in \text{ch}(X_i)$. Parameter $\theta = \{\theta_i\}_{i=1,2,\dots,n}$ is a set of conditional probability distributions, and $\theta_i = P(X_i | \text{pa}(X_i))$ represents conditional probability distributions of X_i when $\text{pa}(X_i)$, the parent set of X_i , is given. Based on the conditional independence hypothesis, the Bayesian network decomposes the joint probability distribution and decomposes into

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{pa}(X_i)), \quad (1)$$

where n is the number of nodes and X_i is the i -th node, which can take discrete and continuous values.

The BN structure learning algorithm based on dynamic programming is developed based on the following idea. If dataset $\mathbf{D} = \{D_1, D_2, \dots, D_N\}$ is given, find a structure in the search space \mathbf{G}^* :

$$\mathbf{G}^* = \arg \max_{\mathbf{G} \in \mathbf{G}_n} \text{score}(\mathbf{G}), \quad (2)$$

where \mathbf{G}_n represents all possible structures of the variable set $V = \{X_1, X_2, \dots, X_n\}$ in DAG search space. $\text{score}(\cdot)$ is a decomposable scoring function. In this article, BIC scoring function is adopted, as shown in the following equation:

$$\text{BIC}(\mathbf{G} | \mathbf{D}) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \theta_{ijk} - \frac{1}{2} \sum_{i=1}^n q_i (r_i - 1) \log N, \quad (3)$$

where q_i indicates the number of all possible parent node sets $\text{pa}(X_i)$ of node X_i , and N_{ijk} is the number of samples when the k -th value of node X_i and the j -th value of its parent node are taken. In $\theta_{ijk} = N_{ijk} / N_{ij}$ ($0 \leq \theta_{ijk} \leq 1$, $\sum_k \theta_{ijk} = 1$, $N_{ij} = \sum_k N_{ijk}$), the second term is a penalty term added to avoid overfitting.

Dynamic programming algorithm recursively traverses DAG space to search the optimal BN structure. Since the structure is acyclic, at least one node of the final structure has no output arc, which is called the leaf node. Assume that the optimal BN structure has a leaf node X and the state transition equation of dynamic programming is

$$\max \text{Score}(\mathbf{V}) = \max_{X \in \mathbf{V}} \left\{ \text{Score BIC}\left(\frac{\mathbf{V}}{X}\right) + \text{Best Score}\left(\frac{X, \mathbf{V}}{X}\right) \right\}, \quad (4)$$

$$\text{Best Score}\left(\frac{X, \mathbf{V}}{X}\right) = \max_{PS \in \mathbf{V} / \{X\}} \text{Local Score}(X | PS). \quad (5)$$

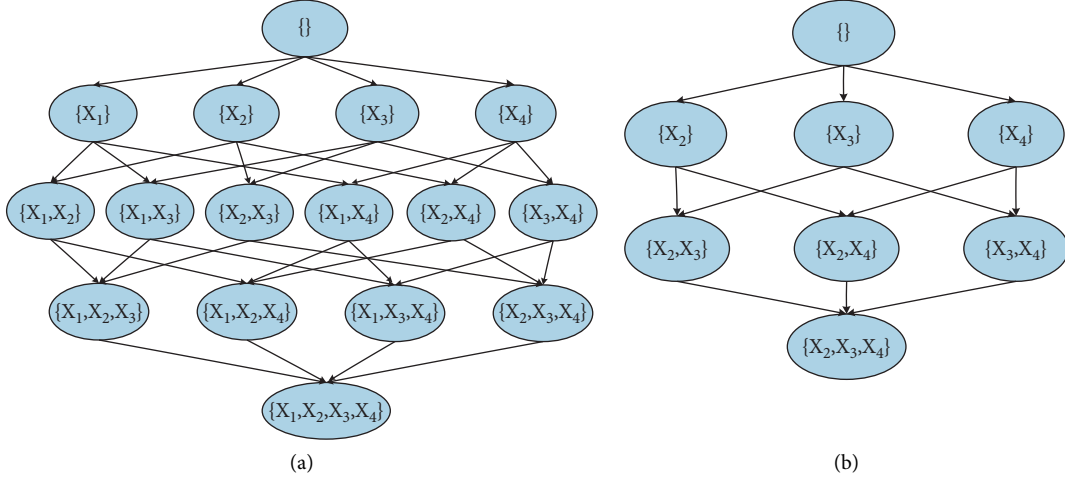
The order graph composed of subsets in \mathbf{V} and the parent graph of each node can represent the whole process of dynamic programming. Figure 1 shows the order graph with the number of nodes $n = 4$ and the parent graph of X_1 . When the DP algorithm calculates from top to bottom, first determine the root node, and then gradually add leaf nodes until the remaining nodes are full sets. Conversely, when the DP algorithm calculates from bottom to top, first determine leaf nodes, and then gradually add root nodes until the remaining nodes are empty sets. In Figure 1, bolded lines indicate that the node order is $X_4 < X_1 < X_3 < X_2$.

It can be seen from Figure 1 that the BNSL algorithm based on dynamic programming needs to calculate and store all the local scores in advance. For the BN with n nodes, any node X_i has 2^{n-1} possible parent sets, requiring $n2^{n-1}$ times of local score calculation in total. Therefore, the time complexity and space complexity of the dynamic programming algorithm are both $O(n2^{n-1})$.

4. Algorithm Idea

The main idea of the proposed algorithm is to use the maximum number of parents obtained with the maximum weight spanning tree and the **CPC** of the node obtained with the MMPC algorithm as the candidate parent set **PS** of the node to lay double constraints on the scoring process based on the dynamic programming algorithm. It further reduces the number of local scoring calculations and the time consumption of the algorithm. In addition, we also propose the S-MMPC algorithm, which uses the symmetry principle to reduce the occurrence of false-positive nodes in **CPC**. Before explaining the detail of this algorithm, the relevant concepts are given first.

Definition 1 (mutual information [40]). The mutual information between two random variables X_i, X_j can be calculated by the following equations:

FIGURE 1: Order graph with $n = 4$ (a) and parent graph of X_1 (b).

$$I(X_i, X_j) = \sum_{i=1}^{r_i} \sum_{j=1}^{r_j} \left\{ P(X_i = x_i, X_j = x_j) \times \lg \frac{P(X_i = x_i, X_j = x_j)}{P(X_i = x_i)P(X_j = x_j)} \right\}, \quad (6)$$

$$I(X_i, X_j) = I(X_j, X_i), \quad (7)$$

where r_i is the number of node states and the value range of X_i is $\{X_{i1}, X_{i2}, X_{i3}, \dots, X_{ir_i}\}$. If $I(X_i, X_j) > 0$, it represents that an undirected edge directly exists between variable X_i and X_j . The larger the $I(X_i, X_j)$ is, the higher the dependence degree between the two variables. If $I(X_i, X_j) = 0$, it represents variable X_i and variable X_j are independent of each other. Equation (7) shows that the mutual information matrix $\mathbf{I}_{n \times n}$ is a symmetric matrix whose diagonal is 0.

Theorem 1 (d-separate and conditional independence). *In BN, if all paths between node X_i and node X_j are blocked by node set \mathbf{Z} , then we can say set \mathbf{Z} d-separates X_i and X_j , or X_i and X_j are independent of each other under the given \mathbf{Z} . It is expressed as $X_i, X_j | \mathbf{Z}$ or $I_D(X_i, X_j | \mathbf{Z})$.*

Definition 2 ($\text{assoc}(X, T | \mathbf{S})$). It indicates the association degree of X and T under the given \mathbf{S} . Its value is determined by G^2 statistics under the null hypothesis of conditional independence. Assume that N^{abc} is the number of times when $X = a$, $T = b$, and $S = c$ in the sample data. Then, G^2 statistical variable is defined as

$$G^2 = 2 \sum_{a,b,c} N^{abc} \ln \left(\frac{N^{abc} N^c}{N^{ac} N^{bc}} \right). \quad (8)$$

Given a significance level α , if the p -value calculated by the G^2 test is less than α , the hypothesis is denied; that is, variable X and variable T are conditionally dependent. Otherwise, they are conditionally independent.

Definition 3 ($\text{Min Assoc}(X, T | \mathbf{Z})$). If all subsets of a given condition set \mathbf{Z} are condition sets, take the minimum value of the association degree of target variables X and T :

$$\text{Min Assoc}(X, T | \mathbf{Z}) = \min_{\mathbf{S} \subseteq \mathbf{Z}} \text{assoc}(X, T | \mathbf{S}). \quad (9)$$

Theorem 2 (false-positive node). *In any subset of the candidate node set $\mathbf{V} \setminus X$, if the conditional independence test of two nodes shows that they are conditional dependent, then the two nodes are adjacent. Otherwise, one node is called the false-positive node of the other.*

4.1. MWST Algorithm. The steps of the MWST algorithm [41] are as follows: first, start from the node set $\mathbf{Y} = \{X_i\}$. Then, find node X_j from the set $\mathbf{V} \setminus \mathbf{Y}$, which has the largest mutual information with any node y in set \mathbf{Y} and uses undirected edges to connect y and X_j . Repeat this operation until $\mathbf{Y} = \mathbf{V}$. At this point, the sum of mutual information between nodes is the largest. Figure 2 shows the structure of Asia network and its MWST structure when data amount $N = 1000$.

It can be seen that there are undirected edges in the tree structure. Therefore, it is impossible to determine the exact parent set and child set of one node. The weight of edges in the tree structure represents the dependency between two variables. Mapped in the BN structure, it shows the relationship between parent nodes and child nodes. In this case, the maximum number of node connections in the tree structure can be considered the maximum number of parent nodes in the network structure. In the Asia network, the maximum number of node connections is 4; that is, the maximum parent number of the network can be set to 4. Then, the candidate parent set of X_6 can be regarded as $\{X_3, X_4, X_7, X_8\}$. It is worth noting that if the number of nodes n is large, the tree structure will lose edges. Therefore, the

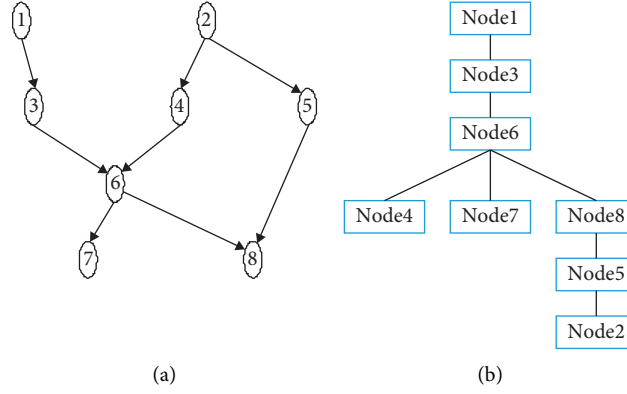


FIGURE 2: Asia network (a) and its MWST structure (b).

maximum number of parent nodes of each node cannot be determined according to the tree structure. For example, if the maximum number of parent nodes of X_3 is 2, and the lost edge happens to the edge between X_3 and a certain node, it will affect the search for the optimal parent set of X_3 . Then, the accuracy of the algorithm cannot be guaranteed.

Based on the above analysis, we use the maximum number of parent nodes obtained from the maximum weight spanning tree structure to prune the parent graph of the DP algorithm horizontally and reduce the upper limit of the parent number from $n-1$ to t , thus decreasing local scoring times. Figure 3 is an example of horizontal pruning of the parent graph by using value t in the Weather network, where the gray sets do not need to be calculated. If $t = 1$, only two layers of scores need to be calculated. If $t = 2$, only three layers of scores need to be calculated. If and only if $t = n$, the constraints are invalid, and the local scoring calculation times of the proposed algorithm are the same as those of the DP algorithm.

In generality, when the network has n nodes, the parent graph of the DP algorithm has n layers. After constraints with value t , $n-t-1$ layers of the parent graph are pruned, and $t+1$ layers are remaining. In this case, the number of local scores to be calculated is $n \sum_{i=0}^t C_{n-1}^i$ ($t \leq m \leq n-1$). After constraints, the local score calculation equation of X is expressed as

$$\max_{PS \in V \setminus \{X\}} \text{Local Score}(X|PS) \Rightarrow \max_{PS \in V \setminus \{X\} \cap |PS| \leq t} \text{Local Score}(X|PS). \quad (10)$$

4.2. S-MMPC Algorithm. Before introducing the S-MMPC algorithm proposed in this article, we would briefly explain the MMPC algorithm. The MMPC algorithm is divided into two stages: in the first stage, it uses the max-min heuristic strategy to make variables enter the **CPC** set of target variable T . In the second stage, it removes the edges that should not enter the **CPC** in the first stage by the following methods: for target variable T , if there is a subset S and a variable X in its **CPC** set that make T and X are conditionally independent with the given S , we can have $\text{Ind}(T, X|S)$. Then, it removes X from the **CPC** of T . But the parent-child node set of the

target node T obtained by the MMPC algorithm may contain false-positive nodes. As shown in Figure 4, to obtain the parent-child node set of the target node T , the nodes B and T are independent if the size of the condition set is 0. Then, delete node B from $\text{Adj}(T)$. However, node T and node C are conditionally independent under condition set $\{A, B\}$. Therefore, deleting node B too early will lead to the failure to test the conditional independence of T and C in subsequent conditional independence tests. The result might be that node C still exists in the parent-child node set of T after the MMPC processing.

To solve this problem, this article proposes the S-MMPC algorithm, which uses the principle of symmetry to improve the MMPC algorithm; that is, two nodes must be in the set of parent and child nodes of each other; otherwise, they are false-positive nodes and can be deleted. As shown in Figure 4, process node T with the MMPC algorithm and we can obtain the parent-child node set, $\text{CPC}(T) = \{A, C\}$. Call MMPC processing on C to get the parent-child node set, $\text{CPC}(C) = \{A, B\}$, $C \in \text{CPC}(T)$ but $T \notin \text{CPC}(C)$, which does not satisfy the symmetric relationship between nodes. Therefore, C is a false-positive node of T , and C should be deleted from the parent-child node set of T . The pseudocode of the S-MMPC algorithm, in which lines 16–20 adopt the symmetry principle to delete false-positive nodes, is shown as follows:

After the constraint of the maximum weight spanning tree, the possible parent set is still $V \setminus X$. Besides, only $PS \in V \setminus X$ and $|PS| \leq t$ are required for the candidate subset of X , which only reduces a small number of scoring calculations. Therefore, this article uses the S-MMPC algorithm to constrain the candidate parent set of nodes. After constraint, the local score calculation equation of X is

$$\max_{PS \in V \setminus \{X\}} \text{Local Score}(X|PS) \Rightarrow \max_{PS \in \text{CPC}(X)} \text{Local Score}(X|PS). \quad (11)$$

According to equation (11), only the following should be met for each node: $|\text{CPC}(X)| \leq n-2$. It means that delete at least one element that does not belong to the candidate parent-child set from the candidate parent set of each node, and then only the local scores of elements in $\text{CPC}(X)$ need to be calculated, thus reducing the score calculations. The number of score calculations at this time is $n2^{|\text{CPC}(X)|}$.

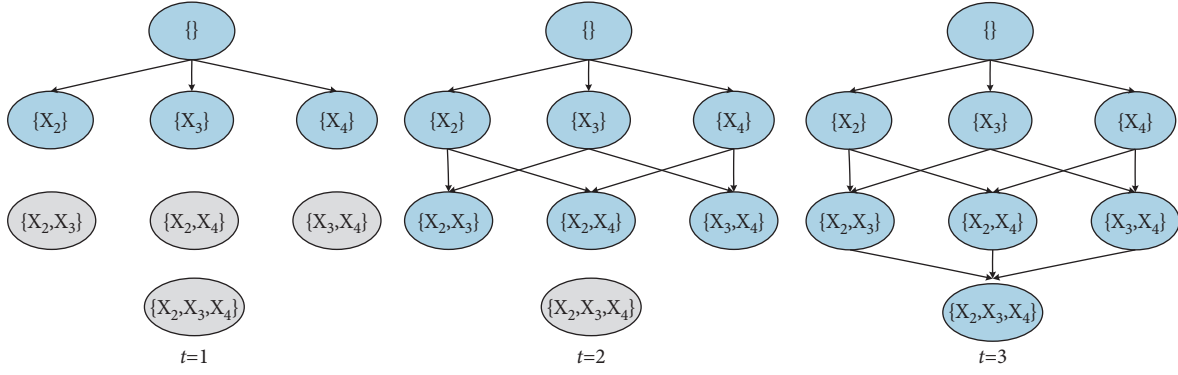


FIGURE 3: Schematic diagram of pruning the parent graph of variable X_1 with value t . (a) $t = 1$. (b) $t = 2$. (c) $t = 3$.

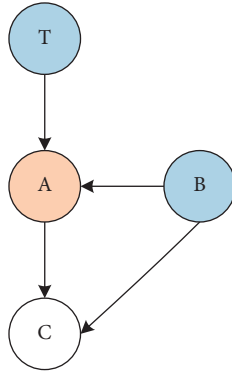


FIGURE 4: Schematic diagram of symmetry principle.

Figure 5 is an example of the Weather network pruning parent graph with $CPC(X_1)$. If $CPC(X_1) = \{X_2, X_3\}$, it means the node sets containing X_4 can be omitted from calculation. They are represented with the gray sets in Figure 5. By comparing Figures 3 and 5, it can be seen that in the Weather network, the DP algorithm needs to calculate local scores 32 times. After the constraint with $t = 2$, local scores need to be calculated 28 times. After the constraint with the S-MMPC algorithm, local scores need to be calculated only 16 times.

4.3. MMPCDP Algorithm. The core idea of the MMPCDP algorithm is to use the maximum number of parent nodes obtained by the maximum weight spanning tree and the candidate parent set obtained by the MMPC algorithm to constrain the dynamic programming process, thereby reducing the search space and improving the efficiency of the algorithm. The block diagram of MMPCDP is shown in Figure 6, and the complete steps of the algorithm are as follows:

Step 1. Use formula (6) to calculate the mutual information matrix between nodes according to the data.

Step 2. Using the size of mutual information as the weight, use the prim algorithm to build the maximum weight spanning tree.

Step 3. Traverse the maximum weight spanning tree to get the maximum number of connections of the node, and use it as the maximum number of parent nodes of the node.

Step 4. Use formula (8) for conditional independence test, the MMPC algorithm improved by the symmetry principle is used to obtain the candidate parent set CPC of the node and use it as the candidate set PS of the node.

Step 5. Recursively traverse the node order space for dynamic programming according to formulas (4), (5), use the maximum number of parent nodes and the candidate set PS to constrain the planning process of dynamic programming, and calculate the structure score according to formula (3).

Step 6. The structure with the highest score is the final BN structure.

Pseudocodes of the MMPCDP algorithm are shown in Algorithm 2. In Algorithm 2, Function3 is used to calculate and store all the local scores that need to be calculated. Lines 1–3 are to obtain the maximum number t of parent nodes of the network by using the tree structure. Line 4 calls Function1 of the S-MMPC algorithm in Section 3.2 and adopts the G^2 test to verify the conditional independence between nodes to get the candidate parent set of nodes. Lines 5–7 call Function4 to calculate the possible parent sets and corresponding scores for each node. In Function4, lines 1–3 are the possible parent set combinations generated by function *nchoosek*. They are stored in set *pa_comb*. Lines 5–13 use double-layer circulation to get the corresponding scores of possible parent sets of all nodes. In line 5, value t is recycled to prune the parent graph horizontally. In line 6, the candidate parent set CPC is recycled to prune the parent graph longitudinally. Line 14 sorts the resulting key-value pairs [*Score*, *cpcodei*] by *score* in descending order, supporting the search of node order graph.

In this article, after biconstraints with the above method, the final local score calculation equation of X is

$$\max_{PS \in V \setminus \{X\}} \text{Local Score}(X|PS) \Rightarrow \max_{PS \in CPC(X) \cap |PS| \leq t} \text{Local Score}(X|PS). \quad (12)$$

Function1 $CPC = s_MMPC(data, ns, n, \alpha, t, I)$
Input: D -Data set, T -Target variable, α -Significance level
Output: CPC-Candidate parent-child node set

- (1) $CPC = \phi$
- (2) while true//Forward-add nodes to CPC
- (3) $\langle F, \text{assoc } F \rangle = \text{Max Min Heuristic}(T, CPC)$
- (4) if $\text{assoc } F \neq 0$
- (5) $CPC = CPC \cup F$
- (6) else
- (7) break
- (8) end if
- (9) end while
- (10) for each variable $X \in CPC$ //Backward-remove nodes from CPC
- (11) if there exists a subset $S \subseteq CPC$ so that $X \perp T | S$
- (12) $CPC = CPC / \{X\}$ //If the disjoint sets of X and T are found in CPC, remove X from the CPC
- (13) end if
- (14) end for
- (15) for all $X \in CPC$
- (16) if $T \notin CPC(X)$
- (17) $CPC = CPC / \{X\}$
- (18) end if
- (19) end for
- (20) return CPC

Function2 $\langle F, \text{assoc } F \rangle = \text{Max Min Heuristic}(T, CPC)$
Input: T -Target variable, CPC-candidate parent-child node set
Output: Node F in the greatest association degree with T and association degree $\text{assoc } F$

- (1) $\text{assoc } F = \max_{X \in V} \text{Min Assoc}(X, T | CPC)$
- (2) $F = \arg \max_{X \in V} \text{Min Assoc}(X, T | CPC)$
- (3) return $F, \text{assoc } F$

ALGORITHM 1: The pseudocode of S-MMPC.

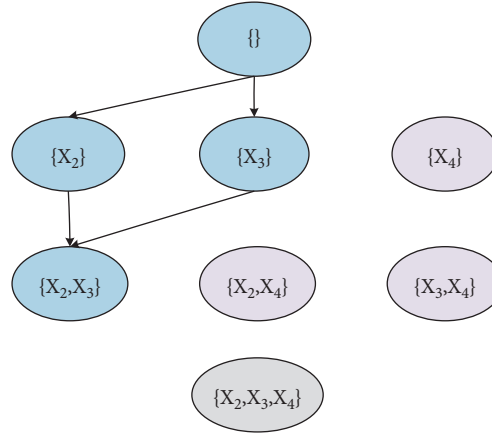
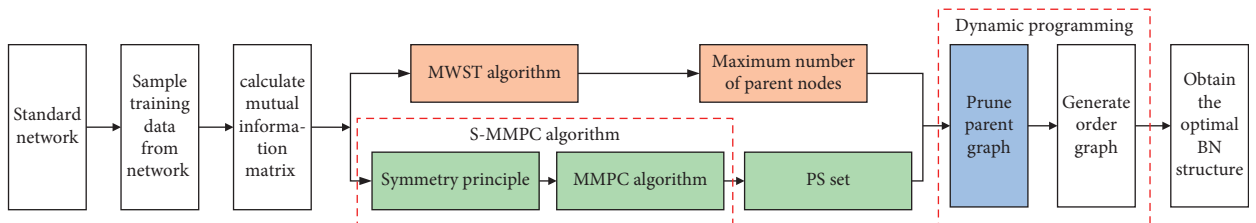
FIGURE 5: Schematic diagram of pruning the parent graph of $(X)_1$ with $CPC(X)_1$.

FIGURE 6: Block diagram of the MMPCDP algorithm.

Function3 [Score Cache cpccode] = calculate scores (data, ns, I)

Input: data-data, ns-node status number, I-Mutual information matrix

Output: scorecache-candidate parent sets and its score, cpccode-decimal encoding of candidate parent set

- (1) $[n, N] = \text{size}(\text{data})$
- (2) $[\text{link}, T1] = \text{prim_link}(I, n)$
- (3) $t = \text{Max_lim_pa}(T1)$
- (4) $CPC = s_MMPC(\text{data}, ns, n, \alpha, t, I)$
- (5) for $X \in V$
- (6) $[\text{score cache}\{xi\}, \text{cpccode}(xi)] = \text{calculates cores}(xi, \text{data}, n, N, CPCi, t)$
- (7) end for

Function4 [score cache{xi}, cpccode(xi)] = calculate scores (xi, data, n, N, CPCi, t)

Output: Score-candidate parent sets and its scores, cpccodei- decimal encoding of candidate parent set

- (1) for $k \leftarrow 0$ to t
- (2) $pa_comb\{k\} = \text{nchoosek}(CPCi, k)$
- (3) end for
- (4) $[\text{Score}, \text{cpccodei}] \leftarrow \emptyset$
- (5) for layer $\leftarrow 0$ to t
- (6) for each node x_i such that $P \in pa_comb\{k\} \& |P| = \text{layer}$
- (7) $\text{Best Score}(X, P) = \max \text{Best Score}(X, P/\{Y\})$
- (8) if $\text{score}(X, P) > \text{Best Score}(X, P)$
- (9) $\text{Best Score}(X, P) \leftarrow \text{score}(X, P)$
- (10) append $[\text{Score}, \text{cpccodei}]$ with $[\text{Score}(X, P), \text{bitnarize}(p)]$
- (11) end if
- (12) end for
- (13) end for
- (14) sort $[\text{Score}, \text{cpccodei}]$ with score in descending
- (15) return $[\text{Score}, \text{cpccodei}]$

ALGORITHM 2: Partial pseudocode of MMPCDP algorithm.

It can be seen that the proposed algorithm reduces the overall score calculation times from $n2^{n-1}$ to $\sum_{j=0}^n \sum_{i=0}^t C_{|CPC\{j\}|}^i (t \leq m \leq n-1)$.

4.4. Algorithm Time Complexity Analysis. The time complexity of the algorithm is measured by the number of executions of the basic operations required during the execution of the algorithm. Assuming that n is the number of network nodes and N is the number of samples, the time complexity of calculating the mutual information of a certain group of nodes is $o(N)$, a total of $n(n-1)/2$ mutual information calculations are required, and the time complexity of calculating the mutual information of the entire network is $o(Nn^2)$. The time complexity of using mutual information to construct the maximum weight spanning tree structure is $o(n^2)$, the time complexity of searching the weight spanning tree to get the maximum parent number of nodes t is $o(1)$, and therefore, the time complexity of searching the tree to obtain the maximum parent number of nodes is $o(n^2(1+N))$; the time complexity of dynamic programming constrained with t and CPC is $o(n2^t)$; therefore, the time complexity of the MMPCDP algorithm is $o(n(2^t + Nn + n))$.

5. Simulation Experiment

5.1. Experiment Conditions. To verify the effectiveness of the MMPCDP algorithm, self-constructed networks with data amount N of 10,000 and different scales (<https://pan.baidu.com/>

[s/1FbrobC5j7y5bVlz1WutY4Q](https://pan.baidu.com/s/1FbrobC5j7y5bVlz1WutY4Q)) are adopted to analyze its performance based on Bayesian toolbox FullBNT-1.0.7 [42]. The software environment is Windows 10 64bit operating system, Intel(R) Core TM) i7-3615QM CPU@ 2.30 GHz processor, and MATLAB R2014a. In order to avoid the randomness of data sampling, which leads to the great difference of t values and affects the algorithm accuracy, the experiment samples 10 groups of data each time, and each group of data runs independently for 10 times to take the average. It means the t values are averaged and rounded up, and the running time is averaged. The self-constructed network parameters are shown in Table 1. n is the number of nodes, E is the number of edges, P indicates the parameter quantity, and λ represents the maximum in-degree.

5.2. Performance of S-MMPC Algorithm. According to Definition 2, significance level α will affect the performance of the MMPC algorithm to a great extent. If α decreases, the result will be stricter, and the obtained $|CPC(T)|$ will be smaller. If α increases, the result will be broader, and the obtained $|CPC(T)|$ will be higher. In this article, sensitivity and specificity are used to evaluate the performance of $|CPC(T)|$ of target variables T obtained by algorithms S-MMPC and MMPC, and to analyze the influence of different values of α on their accuracy. Sensitivity is defined as the number of variables in the CPC set of a target variable, which are correctly identified by the algorithm and belong to the real parent-child node set of the target variable. Specificity is defined as the number of variables that do not belong

TABLE 1: Introduction of network parameters.

n	E	p	λ
10	20	131	3
11	17	267	3
12	24	316	3
13	26	529	3
14	28	465	3
15	30	544	3
16	32	596	3
17	34	388	3
18	36	317	3
19	38	456	3
20	40	768	3
21	42	338	3

to the **CPC** set of the target variable, and variables that belong to the target real parent-child node set correctly identified by the algorithm.

Assume that $\text{num_pc}(X_i)$ represents the number of variables in the parent-child node set of X_i in a standard

network. $\text{mmpc}(X_i)$ and $\text{pc}(X_i)$ represent the parent-child node set of X_i obtained by the algorithm and that of X_i in the standard network, respectively. $L(\sigma)$ indicates the number of nodes that make σ true. The sensitivity and specificity are defined as follows:

$$\begin{aligned} \text{sensitivity} &= \frac{1}{n} \sum_{i=1}^n \frac{L[\text{mmpc}(X_i) \in \text{pc}(X_i)]}{\text{num_pc}(X_i)}, \\ \text{specificity} &= \frac{1}{n} \sum_{i=1}^n \frac{L[\text{mmpc}(X_i) \notin \text{pc}(X_i)]}{n - 1 - \text{num_pc}(X_i)}. \end{aligned} \quad (13)$$

The two statistics together provide relevant measures for evaluating the algorithm performance of S-MMPC and MMPC, and either statistic alone is not sufficient. In this article, Euclidean distance is used as a combination measure to characterize the approximate degree between the sensitivity and specificity of the algorithm and the optimal sensitivity and specificity:

$$d = \sqrt{(1 - \text{sensitivity})^2 + (1 - \text{specificity})^2}. \quad (14)$$

The smaller the distance d is, the closer the output of the algorithm is to the true parent-child node set.

In this article, Asia network, Sachs network, Child network, and Insurance network are selected from BN repository to evaluate the performance of the S-MMPC algorithm, and $N=10,000$. Figure 7 shows the sensitivity, specificity, and Euclidean distance of S-MMPC and MMPC under four different values of α .

In Figure 7, it can be seen that in the above four standard networks of different scales, when the amount of data is constant, the Euclidean distance increases with the significance level α . When the four networks have the same significance level α , the Euclidean distance of the proposed S-MMPC algorithm is smaller than that of the MMPC algorithm, which means the S-MMPC algorithm has a better performance. It can ensure that the output parent-child set of nodes is close to the real one and reduce the error rate of longitudinal pruning of the parent graph by the S-MMPC

algorithm. The reason for the performance improvement is that the S-MMPC algorithm can improve the specificity by deleting false-positive nodes according to the symmetry principle with the same sensitivity as the MMPC algorithm, thus reducing Euclidean distance.

5.3. Performance of MMPCDP Algorithm. In order to avoid traversing equivalent structures, we evaluate the accuracy of the algorithm by the percentage of the error between the network structure score calculated by the algorithm and the original network structure score. The calculation equation is

$$D = \left| \frac{\text{score}(\mathbf{G}') - \text{score}(\mathbf{G})}{\text{score}(\mathbf{G})} \right| \times 100\%, \quad (15)$$

where \mathbf{G}' is the network structure learned by the proposed algorithm, \mathbf{G} represents the original network structure, and function score uses BIC. Equation (15) shows that the smaller the scoring error percentage, the higher the accuracy of the algorithm.

5.3.1. Analysis of Transverse Pruning. Different networks have different generated spanning tree structures and t values. This section compares the size of the maximum in-degree λ of the network and t obtained with MWST. Then, it judges whether the accuracy of the DP algorithm will be influenced by horizontally pruning the parent graph with MWST. The statistical results are shown in Figure 8.

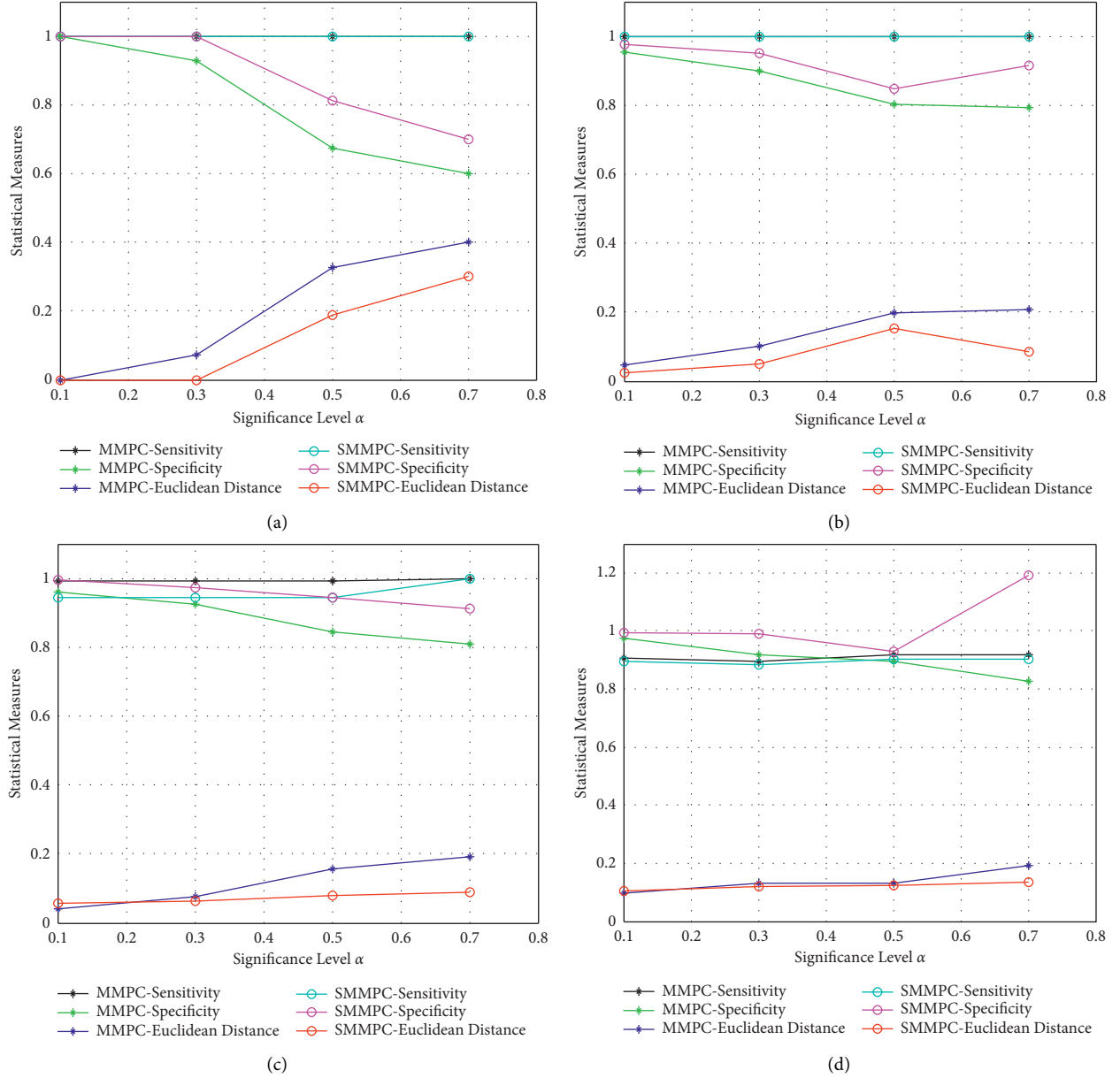


FIGURE 7: Sensitivity, specificity, and Euclidean distance curves of the two algorithms under different values α . (a) Asia. (b) Sachs. (c) Child. (d) Insurance.

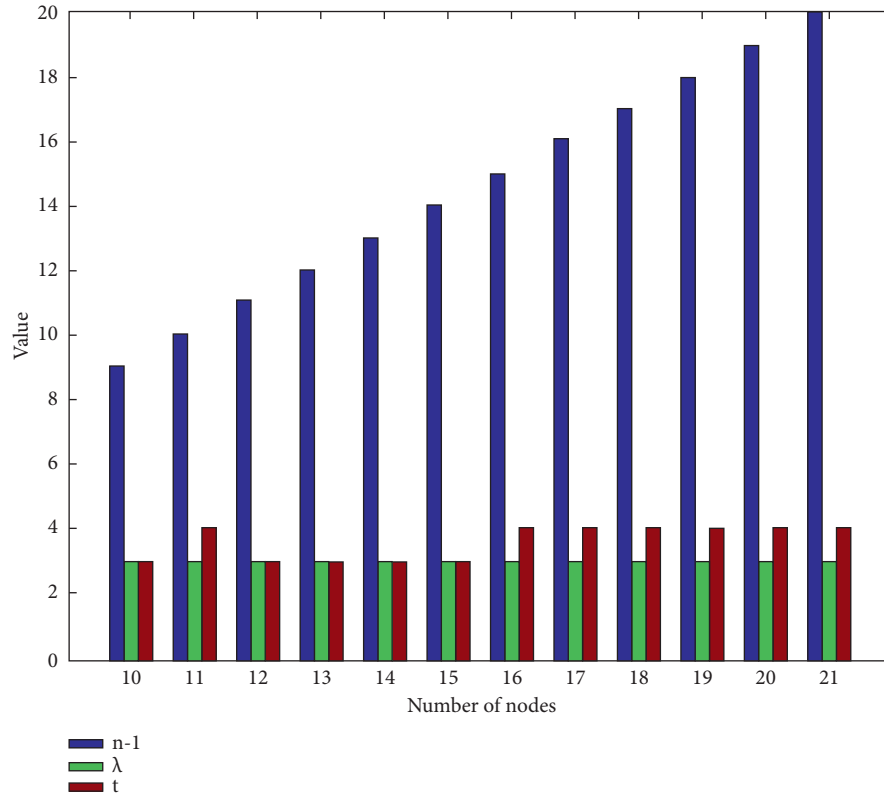
It can be seen that by comparing the values of λ and t , we have $\lambda \leq t \leq n - 1$. It means each parent graph after horizontal pruning with value t must contain the optimal parent set of the node, and the algorithm accuracy will not be affected. Therefore, the accuracy of the proposed algorithm depends on longitudinal pruning.

5.3.2. Analysis of Longitudinal Pruning. In this article, the CPC obtained by the S-MMPC algorithm is used as the candidate parent set of nodes, and the parent graph is pruned longitudinally. However, according to Definition 2, it can be seen that significance level α will affect the

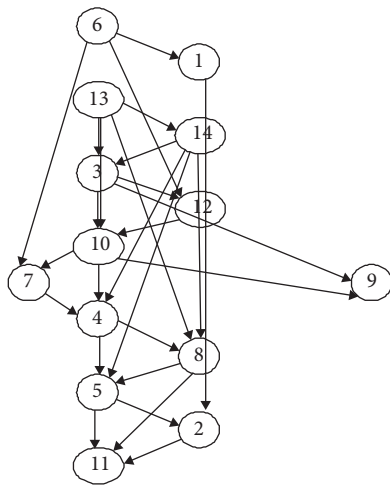
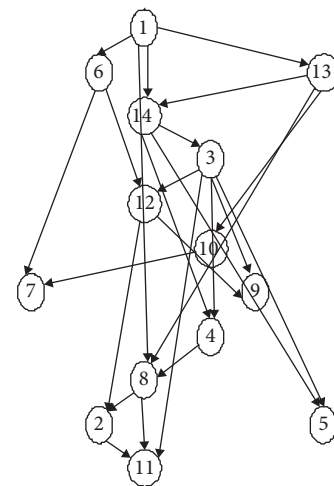
performance of the S-MMPC algorithm to a great extent. A detailed analysis is shown in Table 2.

Therefore, this section takes a network with $n = 14$ as an example (Figure 9). Then, it analyzes the influence of different values of α on the accuracy of the MMPCDP algorithm. Figure 10 shows the network structure obtained by the MMPCDP algorithm when $\alpha = 0.5$.

It can be seen from Figure 11 that the running time of the MMPCDP algorithm increases rapidly with the significance level α . Score error percentage decreases as α tends to 1. It finally stabilizes and reaches 0. Therefore, accuracy and time are two indexes that cannot be achieved simultaneously in the algorithm. To strike a balance between the accuracy and time

FIGURE 8: t and λ of different networks.TABLE 2: The effect of parameters α on algorithm performance.

Condition	Result	$ CPC(T) $	Accuracy of MMPCDP	Time consumption of MMPCDP
$\alpha \downarrow$	Stricter	\downarrow	\downarrow	\downarrow
$\alpha \uparrow$	Broader	\uparrow	\uparrow	\uparrow

FIGURE 9: Network structure diagram with $(n) = 14$.FIGURE 10: Network structure obtained $\alpha = 0.5$.

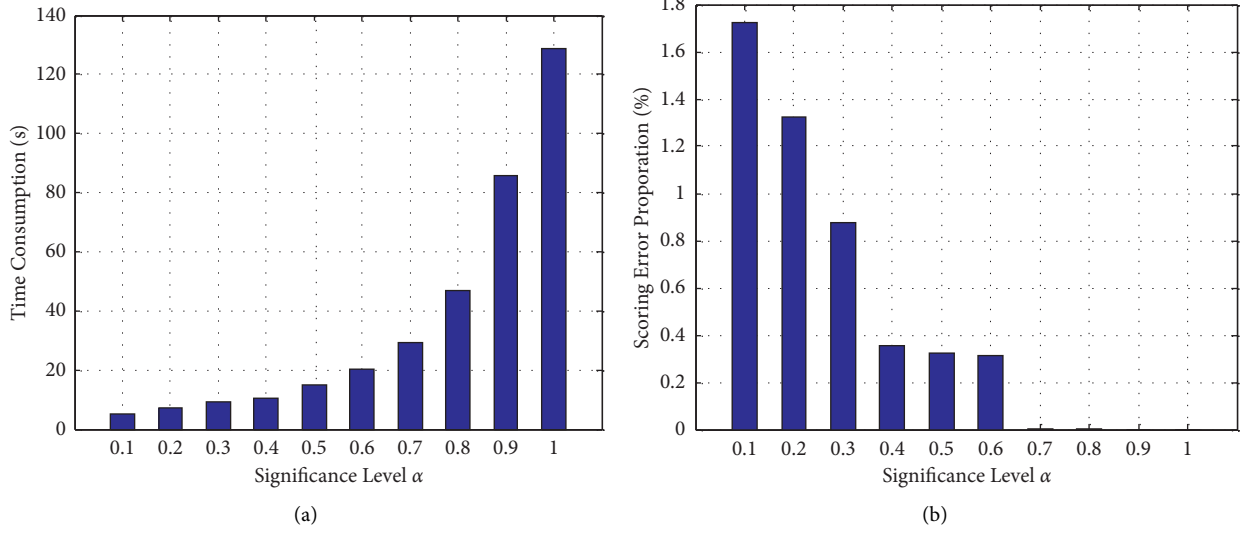


FIGURE 11: Influence of significance level α on indexes of the proposed algorithm. (a) Time consumption of different α . (b) Scoring error proportion of different α .

of the algorithm, Section 5.3.3 takes $\alpha = 0.5$ to measure the performance of the proposed algorithm.

5.3.3. Analysis of Time Consumption and Accuracy. Set $N = 10000$, $\alpha = 0.5$, and apply the proposed algorithm, standard DP algorithm, SMDP algorithm, MEDP algorithm, and DPCMB algorithm to the network in Section 5.1. Here, the importance threshold ε of the DPCMB algorithm is set to 0.50. Then, compare the algorithms' running time to reflect time consumption, and compare their scoring error percentages to show accuracy. The results are given in Table 3, where “—” indicates that the storage space required by the algorithm exceeds the memory of our computer. Figure 12 shows the time consumption ratio of SMDP, MEDP, DPCMB, and MMPCDP based on the time consumption of the DP algorithm.

For the part marked in bold, such as $n = 13, 14, 16$, and 21, the accuracy of the MMPCDP algorithm proposed in this article has become lower, and it has not reached the situation that the accuracy of the SMDP algorithm is consistent and close to 0, and then research separately, conduct experiments, and gradually increase the value of α to make the accuracy approach the accuracy of the SMDP algorithm; the results of various indicators are shown in Table 4.

Compared with the other four algorithms, the MMPCDP algorithm not only ensures the accuracy, but also reduces the time consumption. In Figure 12, the average running time of the five algorithms is roughly ranked as follows: $\text{MMPCDP} \leq \text{DPCMB} \leq \text{MEDP} \leq \text{SMDP} \leq \text{DP}$. With the increase of n , the effectiveness of the constraints becomes more and more obvious. As shown in Table 4, by increasing the importance threshold α , the score error percentage of the MMPCDP algorithm is mostly increased to the same point as the score error percentage obtained by the SMDP algorithm; that is, when α gradually approaches 1, the accuracy of the MMPCDP algorithm approaches the SMDP algorithm and eventually stabilizes.

5.4. Discussion. From the perspective of local scoring times, constructing P-caches storage structure with SMDP algorithm still needs $n2^{n-1}$ times. The MEDP algorithm uses MDL scoring function to reduce the maximum number of parent nodes from $n-1$ to m . Its required local scoring times are $n \sum_{i=0}^m C_{n-1}^i$ ($m = \log(2N/\log N)$). The sample size will affect the final structure accuracy. The DPCMB algorithm uses Markov blanket to constrain the calculation process, and the required number of local scoring times is $\sum_{X \in V} 2^{|MB(X)|}$. In this article, the proposed algorithm uses double constraints to reduce the total score calculation times from $n2^{n-1}$ to $\sum_{j=0}^n \sum_{i=0}^t C_{|CPC\{j\}|}^i$ ($t \leq m \leq n-1$). It can be seen that our algorithm reduces its time consumption and complexity by reducing the number of scoring times. The algorithm accuracy is subject to the significance level α , but not affected by the horizontal pruning with t .

6. UAV Intelligent Decision-Making Application Based on MMPCDP

6.1. Air Combat Scene Description. First, give the mission scenario: there are multiple enemy air targets invading, and our side dispatches manned aircraft and drones for air combat. Due to strong electromagnetic interference from the enemy, the communication between the manned aircraft and the drone is interrupted. In order to ensure the safety of the manned aircraft, we decided to withdraw the manned aircraft and let the unmanned aerial vehicle (UAV) conduct autonomous operations to complete the task of air combat. At this time, when a UAV faces multiple enemy targets, it has to choose one target to attack, so it needs to make an attack decision. The task scenario is shown in Figure 13:

Different UAVs are equipped with different sensors, so the situation information that can be obtained is different. This article assumes that UAV can acquire target information

TABLE 3: Time consumption and accuracy of five algorithms

n	Running time (s)					Scoring error percentage (%)				
	DP	SMDP	MEDP	DPCMB	MMPCDP	DP	SMDP	MEDP	DPCMB	MMPCDP
10	15	2	2	1	2	1.0367e-14	1.6415e-14	1.4829e-13	4.1813e-13	1.0354e-13
11	24	4	4	2	3	2.1546e-12	2.2428e-12	2.7766e-12	2.8182e-12	2.9145e-12
12	99	24	11	5	9	5.1105e-13	1.1126e-12	3.2659e-12	0.3510	2.3863e-12
13	262	115	26	12	12	5.9007e-13	1.0534e-12	1.2339e-12	2.1302e-12	0.6110
14	782	253	60	19	15	1.4377e-13	2.9742e-13	3.0249e-13	0.3085	0.3262
15	3128	638	148	27	18	2.2582e-14	7.0055e-14	7.9981e-14	0.0208	7.0080e-14
16	7658	2534	303	41	23	1.5372e-13	1.7624e-13	1.6863e-13	1.2875e-12	0.2656
17	57770	9326	621	103	38	1.0081e-13	1.0435e-13	1.2154e-12	1.0734e-12	1.0307e-12
18	98492	24200	1635	561	45	2.9685e-14	4.7709e-14	8.0055e-14	0.0163	1.2658e-13
19	—	—	5302	734	67	—	—	1.3837e-12	2.0963e-12	1.7139e-12
20	—	—	18525	5447	103	—	—	1.1327e-13	1.3587e-12	1.2378e-12
21	—	—	93363	10328	153	—	—	1.9672e-14	0.6803	0.6036

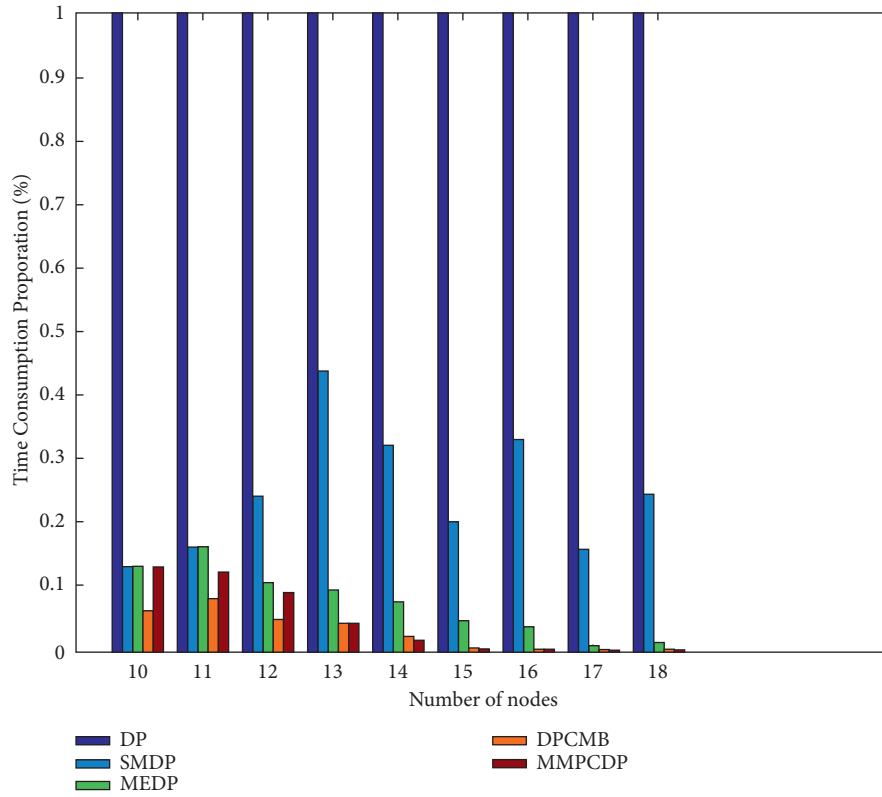


FIGURE 12: running time of the five algorithms.

TABLE 4: Time consumption and accuracy of the MMPCDP algorithm of the network bold marked in Table 3.

n	Running time (s)	Scoring error percentage	α
13	19	2.0317e-12	0.6
14	30	3.1261e-13	0.7
16	41	1.1605e-12	0.75
21	524	2.0731e-12	0.85

of the following: height (H), distance (Dis), velocity (V), angle (A), radiation (R), radar reflection area (RCS), track state (TS), warning information (WI), type (T), intention (I),

capacity (C), importance (Im), and task point (TP), as well as virtual node threat value(TV) and decision variable(D). The following is the discretization of the above variables:

6.2. Intelligent Decision-Making Structure Construction.

This article uses a simulation platform to simulate the combat process to obtain data, and the data are processed according to the discretization method shown in Table 5. The portion of the processed data is shown in Table 6.

Then, use the MMPCDP algorithm proposed in this article to learn the Bayesian network structure, thereby constructing a

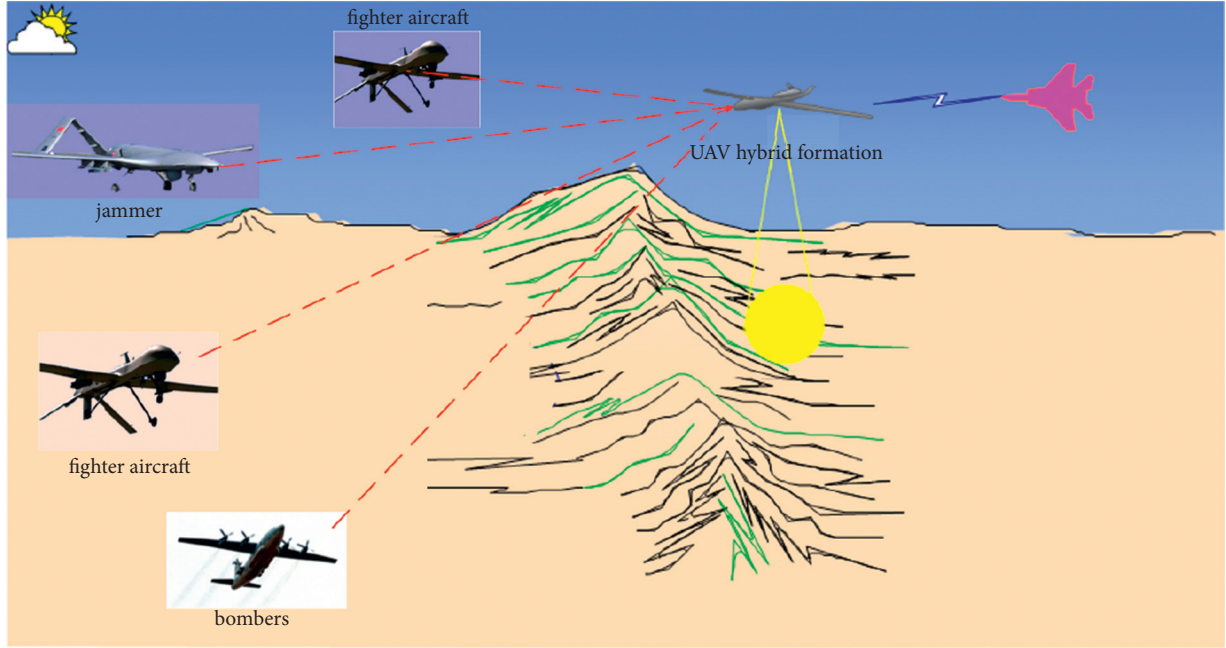


FIGURE 13: Schematic diagram of air combat missions.

TABLE 5: Discretization of the target information variables.

Node number	Target information	Discretization
1	Decision	$\{1,2\} = \{ \text{Don't attack, attack} \}$
2	Threat value	$\{1,2\} = \{ \text{low, medium, high} \}$
3	Type	$\{1,2,3\} = \{ \text{jammer, bombers, fighter aircraft} \}$
4	Intention	$\{1,2\} = \{ \text{interference, attack} \}$
5	Capacity	$\{1,2,3\} = \{ \text{low, medium, high} \}$
6	Importance	$\{1,2,3\} = \{ \text{low, medium, high} \}$
7	Task point	$\{1,2\} = \{ \text{attack the machine, not attack the machine} \}$
8	Height	$\{1,2\} = \{ \text{Height} \geq 3Km, \text{Height} < 3Km \}$
9	Distance	$\{1,2,3\} = \{ \text{Distance} < 20Km, 20Km \leq \text{Distance} < 50Km, 50Km \leq \text{Distance} \}$
10	Velocity	$\{1,2,3\} = \{ \text{Velocity} < 1Ma, 1Ma \leq \text{Velocity} < 2Ma, 2Ma \leq \text{Velocity} \}$
11	Angle	$\{1,2,3\} = \{ \text{Angle} < \pi/3, \pi/3 \leq A \leq 2\pi/3, 2\pi/3 < A \}$
12	Radiation state	$\{1,2\} = \{ \text{close, open} \}$
13	RCS	$\{1,2\} = \{ \text{small reflection area, large reflection area} \}$
14	Track	$\{1,2\} = \{ \text{no, yes} \}$
15	Warning information	$\{1,2\} = \{ \text{no, yes} \}$

TABLE 6: Portion of the processed data.

	D	TV	Ty	I	C	Im	TP	H	Dis	V	A	R	RCS	TS	WI
1	1	1	1	1	2	1	2	2	1	2	3	1	1	2	1
2	1	2	2	2	1	2	2	1	3	1	2	2	1	2	1
3	2	2	3	2	2	3	1	2	1	2	1	1	2	1	2
4	2	3	1	1	3	2	1	2	1	2	2	2	1	1	2
5	2	3	2	2	3	3	1	1	2	2	2	1	2	2	1
6	2	1	1	1	1	3	1	2	3	1	3	2	1	2	1
7	2	2	3	2	3	2	1	2	1	3	2	1	1	2	2
8	1	1	2	1	2	2	2	2	3	2	3	2	2	1	1
9	1	1	3	1	1	3	2	2	1	3	2	1	2	2	1
10	2	2	2	2	3	1	2	2	2	2	1	2	1	1	1

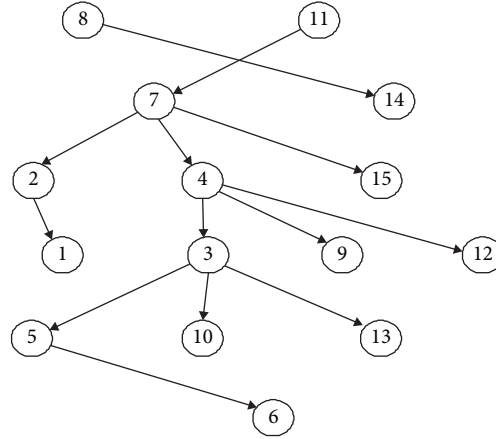


FIGURE 14: The BN structure obtained by expert knowledge.

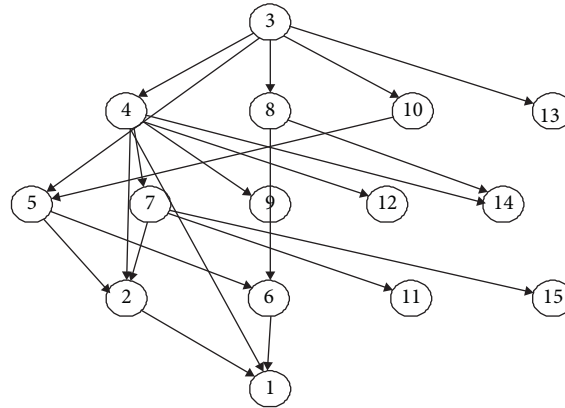


FIGURE 15: The BN structure obtained by the algorithm in this article.

structure for UAV intelligent decision-making. Figure 14 is the BN structure constructed by expert knowledge. The final structure is shown in Figure 15. As the method of dynamic programming is adopted, this structure is the global optimal solution of the data.

By comparing the two structures, it can be found that while retaining expert knowledge, Figure 15 excavates more variable information, which is more in line with the objective facts. For example, the parent node of the decision node increases the target intention and the threat value in addition to the threat value included in the expert knowledge; the parent node of the target threat value node, in addition to the target task orientation of expert knowledge, adds the t target intention and target ability. From this, we can see the practical feasibility of the algorithm in this article.

7. Conclusion

DP algorithm is an accurate algorithm for learning BN structure, but it has high complexity, which limits its application. Therefore, this article proposes the dynamic programming structure learning algorithm of Bayesian network integrating MWST and improved MMPC. This algorithm uses mutual information to build the MWST and takes the maximum number of node connections in the MWST structure as

the maximum number t of parent nodes of the network. The **CPC** set obtained by the S-MMPC algorithm is used as the candidate parent set PS of nodes. By pruning the parent graph horizontally and longitudinally, the parent sets that need to be visited are reduced, thus reducing local scoring calculation times, the memory of scores, and the running time of the algorithm. Compared with the SMDP algorithm, the MMPCDP algorithm uses constraints to reduce the time consumption of the DP algorithm, whereas the SMDP algorithm uses a new storage structure; compared with the MEDP algorithm and DPCMB algorithm, this article has the same improvement ideas as these two algorithms, both by reducing local scores to reduce the time consumption of the DP algorithm, and the MMPCDP algorithm requires the least number of local scores by theoretical analysis. And simulation experiments show that when an appropriate significance level α is selected, the MMPCDP algorithm can reduce scoring times and running time while ensuring accuracy. If and only if $t = |\mathbf{CPC}| = n-1$, the constraints are invalid. In future work, we will study how to effectively adjust the value of α effectively to meet the accuracy requirements.

Data Availability

Data used in the experiments are synthetically generated from the networks.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful to Dr. Zhi-gao Guo for helpful discussions. This work was supported by the National Key Laboratory Fund (CEMEE2020Z0202B), Equipment Development Fund (80902010302), and the Shanxi Science Foundation (2020JQ-816 and 20JK0680).

References

- [1] P. Z. Zhao and Y. Y. Cui, "A new method of mobile ad hoc network routing based on greed forwarding improvement strategy," *IEEE Access*, vol. 7, no. 1, pp. 158514–158524, 2019.
- [2] C. Chen and Y. Y. Cui, "New method of energy efficient subcarrier allocation based on evolutionary game theory," *Mobile Networks and Applications*, vol. 26, no. 2, pp. 523–536, 2021.
- [3] C. L. Gong, D. Zhang, T. Zhang, J. Zhang, and M. Piao, "A new algorithm of clustering AODV based on edge computing strategy in IOV," *Wireless Networks*, vol. 27, no. 4, pp. 2891–2908, 2021.
- [4] D. Zhang, H. Ge, T. Zhang, Y. Cui, X. Liu, and G. Mao, "New multi-hop clustering algorithm for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1517–1530, 2019.
- [5] D. Zhang, G. Li, K. Zheng, X. Ming, and Z.-H. Pan, "An energy-balanced routing method based on forward-aware factor for wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 766–773, 2014.
- [6] A. Pramod, T. Krishnaprasad, and M. Surendra, "Understanding city traffic dynamics utilizing sensor and textual observations," in *Proceedings of the 30th Annual AAAI Conference on Artificial Intelligence*, pp. 3793–3799, AAAI Press, Phoenix, AZ, USA, February 2016.
- [7] X. H. Liu, "A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm," *Cluster Computing*, vol. 24, no. 1, pp. 1–15, 2021.
- [8] T. Zhang, D.-g. Zhang, H.-r. Yan, J.-n. Qiu, and J.-x. Gao, "A new method of data missing estimation with FNN-based tensor heterogeneous ensemble learning for internet of vehicle," *Neurocomputing*, vol. 420, no. 1, pp. 98–110, 2021.
- [9] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Elsevier, Amsterdam, Netherlands, 1988.
- [10] S. Liu, D. Zhang, X. Liu, T. Zhang, and H. Wu, "Adaptive repair algorithm for TORA routing protocol based on flood control strategy," *Computer Communications*, vol. 151, no. 1, pp. 437–448, 2020.
- [11] Y. Y. Cui, "Novel method of mobile edge computation offloading based on evolutionary game strategy for IoT devices," *AEU-International Journal of Electronics and Communications*, vol. 118, no. 5, pp. 1–13, 2020.
- [12] S. McLachlan, K. Dube, G. A. Hitman, N. E. Fenton, and E. Kyrimi, "Bayesian networks in healthcare: distribution by medical condition," *Artificial Intelligence in Medicine*, vol. 107, Article ID 101912, 2020.
- [13] X. Zhang and S. Mahadevan, "Bayesian network modeling of accident investigation reports for aviation safety assessment," *Reliability Engineering & System Safety*, vol. 209, no. 9, pp. 1–19, 2020.
- [14] B. Sun, Y. Li, Z. Wang, D. Yang, Y. Ren, and Q. Feng, "A combined physics of failure and bayesian network reliability analysis method for complex electronic systems," *Process Safety and Environmental Protection*, vol. 148, no. 3, pp. 698–710, 2021.
- [15] R. H. Di, X. G. Gao, and Z. G. Guo, "Small data set BN modeling method and its application in threat assessment," *Chinese Journal of Electronics*, vol. 44, no. 6, pp. 1504–1511, 2016.
- [16] D. Koller and N. Friedman, *Probabilistic Graphical Models*, MIT Press, Cambridge, MA, USA, 2009.
- [17] H. R. Liu, M. T. Sun, and L. Li, "A bayesian network structure algorithm based on ant Colony node optimization," *Journal of Scientific Instruments*, vol. 38, no. 1, pp. 143–150, 2017.
- [18] P. Larrafiaga, M. Poza, Y. Yurramendi, H. Murga, and C. M. H. Kuijpers, "Structure learning of bayesian networks by genetic algorithms: a performance analysis of control parameters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, pp. 912–926, 1996.
- [19] X. Q. Liu, "Structure learning of bayesian networks by continuous particle swarm optimization algorithms," *Journal of Statistical Computation and Simulation*, vol. 88, no. 7, pp. 1528–1556, 2018.
- [20] T. Jaakkola, D. Sontag, and A. Globerson, "Learning bayesian network structure using LP relaxations," *Journal of Machine Learning Research*, vol. 9, pp. 358–365, 2010.
- [21] C. Yuan and B. Malone, "Learning optimal bayesian networks: a shortest path perspective," *Journal of Artificial Intelligence Research*, vol. 48, no. 1, pp. 23–65, 2014.
- [22] M. Koivisto and K. Sood, "Exact bayesian structure discovery in bayesian networks," *Journal of Machine Learning Research*, vol. 5, no. 5, pp. 549–573, 2004.
- [23] X.-H. Liu, D.-G. Zhang, H.-R. Yan, Y.-y. Cui, and L. Chen, "A new algorithm of the best path selection based on machine learning," *IEEE Access*, vol. 7, no. 1, pp. 126913–126928, 2019.
- [24] S. Liu, D.-G. Zhang, X.-h. Liu et al., "Dynamic analysis for the average shortest path length of mobile ad hoc networks under random failure scenarios," *IEEE Access*, vol. 7, pp. 21343–21358, 2019.
- [25] L. Chen and J. Zhang, "A multi-path routing protocol based on link lifetime and energy consumption prediction for mobile edge computing," *IEEE Access*, vol. 8, no. 1, pp. 69058–69071, 2020.
- [26] T. Silander and P. Myllymaki, "A simple approach for finding the globally optimal Bayesian network structure," in *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pp. 445–452, AUAI Press, Cambridge, MA, USA, July 2006.
- [27] B. Malone, C. H. Yuan, and E. A. Hansen, "Memory-efficient dynamic programming for learning optimal bayesian networks," in *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pp. 1057–1062, AAAI Press, San Francisco, CA, USA, August 2011.
- [28] A. P. Singh and A. W. Moore, *Finding Optimal Bayesian Networks by Dynamic Programming*, Carnegie Mellon University, Pittsburgh, PA, USA, 2005.
- [29] S. M. Ye, *Bayesian Network Structure Learning with a Priori and its Application in Intelligent Decision-Making*, Northwestern Polytechnical University, Xi'an, China, 2018.
- [30] X. Y. Tan, X. G. Gao, and C. C. He, "Learning optimal bayesian network structure constrained with Markov blanket," *Acta Electronica Sinica*, vol. 47, no. 9, pp. 1898–1904, 2019.

- [31] Y. Yu, X. G. Gao, and Z. G. Guo, "Finding optimal Bayesian networks by a layered learning method," *Systems Engineering and Electronics*, vol. 30, no. 5, pp. 946–958, 2019.
- [32] S. Behjati and H. Beigy, "Improved K2 algorithm for Bayesian network structure learning," *Engineering Applications of Artificial Intelligence*, vol. 91, pp. 103617.1–103617.12, 2020.
- [33] R. H. Xu, S. H. Liu, Q. W. Zhang, Z. Yang, and J. Liu, "PEWOBS: an efficient Bayesian network learning approach based on permutation and extensible ordering-based search," *Future Generation Computer Systems*, vol. 128, pp. 505–520, 2021.
- [34] Z. Wang, X. Gao, X. Tan, and X. Liu, "Determining the direction of the local search in topological ordering space for Bayesian network structure learning," *Knowledge-Based Systems*, vol. 234, Article ID 107566, 2021.
- [35] X. G. Gao, C. F. Wang, and R. H. Di, "An improved K-means block learning algorithm for learning the optimal structure of sparse BN," *IEEE/CAA Journal of Automatica Sinica*, vol. 46, no. 5, pp. 923–933, 2020.
- [36] C. Yuan, B. Malone, and X. Wu, "Learning optimal Bayesian networks using A* search," in *Proceedings of the International Joint Conference on Artificial Intelligence*, AAAI Press, Barcelona, Spain, July 2011.
- [37] X. H. Liu, "Novel best path selection approach based on hybrid improved A* algorithm and reinforcement learning," *Applied Intelligence*, vol. 51, no. 9, pp. 1–15, 2021.
- [38] X. Tan, X. Gao, Z. Wang, and C. He, "Bidirectional heuristic search to find the optimal Bayesian network structure," *Neurocomputing*, vol. 426, pp. 35–46, 2021.
- [39] Z. Wang, X. Gao, Y. Yang, X. Tan, and D. Chen, "Learning Bayesian networks based on order graph with ancestral constraints," *Knowledge-Based Systems*, vol. 211, Article ID 106515, 2021.
- [40] R. Koopman and S. Wang, "Mutual information based labelling and comparing clusters," *Scientometrics*, vol. 111, no. 2, pp. 1157–1167, 2017.
- [41] C. Chow and C. Liu, "Approximating discrete probability distributions with causal dependence trees," in *Proceedings of the 2010 International Symposium on Information Theory and its Applications*, pp. 100–105, IEEE press, Taichung, Taiwan, October 2010.
- [42] P. K. Murphy, "The bayes net toolbox for MATLAB," *Computer Science and Statistics*, vol. 33, pp. 95–110, 2001.