

## Research Article

# A Mobile Bayesian Network Structure Learning Method Using Genetic Incremental K2 Algorithm and Random Attribute Order Technology

Ying Xiao,<sup>1</sup> Deyan Wang <sup>1,2</sup> and Ya Gao<sup>1</sup>

<sup>1</sup>School of Internet of Things Technology, Wuxi Institute of Technology, Wuxi, Jiangsu, China

<sup>2</sup>Management & Science University, Shah Alam, Malaysia

Correspondence should be addressed to Deyan Wang; wangdy@wxit.edu.cn

Received 22 September 2021; Revised 11 October 2021; Accepted 11 October 2021; Published 19 October 2021

Academic Editor: Tongguang Ni

Copyright © 2021 Ying Xiao 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.

The application of existing datasets to construct a probabilistic network has always been the primary research focus for mobile Bayesian networks, particularly when the dataset size is large. In this study, we improve the K2 algorithm. First, we relax the K2 algorithm requirements for node order and generate the node order randomly to obtain the best result in multiple random node order. Second, a genetic incremental K2 learning method is used to learn the Bayesian network structure. The training dataset is divided into two groups, and the standard K2 algorithm is used to find the optimal value for the first set of training data; simultaneously, three similar suboptimal values are recorded. To avoid falling into the local optimum, these four optimal values are mutated into a new genetic optimal value. When the second set of training data is used, only the best Bayesian network structure within the five abovementioned optimal values is identified. The experimental results indicate that the genetic incremental K2 algorithm based on random attribute order achieves higher computational efficiency and accuracy than the standard algorithm. The new algorithm is especially suitable for building Bayesian network structures in cases where the dataset and number of nodes are large.

## 1. Introduction

The effective expression of uncertain knowledge is an important content of knowledge intelligent learning. In this research field, Bayesian network has always been the focus of attention. Bayesian network is a probability graph model. It represents the dependency relationship between a group of random variables through a directed acyclic graph. The conditional probability table (CPT) formed by each variable represents the probability relationship between variables [1]. It has strong uncertainty reasoning ability, which can realize top-down prediction analysis and bottom-up diagnostic inference [2]. Machine learning technologies have become more and more important in many applications, such as medicine [3], e-commerce [4, 5], transportation [6], and image denoising [7]. As one of the machine learning technologies, Bayesian network is widely used in many

fields, such as machine vision, biomedicine, classification, fault diagnosis, prediction, natural language processing, and data mining [8].

The Bayesian network is primarily based on Bayesian network learning, which is divided into two steps: structural learning and parameter learning. Structure learning is to obtain a directed acyclic graph that can represent attribute dependencies based on training data and a priori knowledge. Parameter learning is to obtain the conditional probability of each node based on the directed acyclic graph. It is usually called conditional probability table. In these two learning, structural learning is more difficult, and it is also a research hotspot. It mainly focuses on how to avoid falling into local optimization and find the best structure when there are many attributes and few sample data [9, 10]. The Bayesian network structure learning algorithm can be divided into

a method based on scoring and searching, the conditional independence test, and a hybrid of the two methods.

- (1) Method based on scoring and searching: it uses the scoring function to measure the matching degree between the Bayesian network structure and the training sample set. After defining the scoring function, apply the search strategy to find the network structure with the highest score. K2 algorithm is the common one [11]. Due to the constraint of node order, K2 algorithm can effectively avoid the problem of likelihood equivalence and is better than most classical algorithms in running speed and accuracy. However, in most cases, the node order is unknown and usually needs to be determined according to expert knowledge. The difference of expert knowledge is large, which cannot ensure objectivity and accuracy, and it is difficult to achieve when there are a large number of nodes. Therefore, researchers have put forward many solutions. It is proposed that conditional frequency is used to determine the node ranking of K2 algorithm. The algorithm does not need complex search strategy and effectively reduces the time complexity. However, it has high requirements for the quality of data set and is not easy to obtain the accurate model [10]. Combining the maximum spanning tree and ant colony algorithm, the MUST-ACO-K2 (MAK) algorithm is proposed to search the node order, but the algorithm needs to substitute the obtained node order into K2 algorithm to get the network structure before scoring, resulting that its running time is too long [10].
- (2) Method based on conditional independence test: this method abstracts the learning process of the Bayesian network structure as the process of discovering a set of variables hidden in the network structure that satisfy the independence condition test. Spirtes proposed the SGS algorithm in 1989 [12], which uses conditional independence to test the existence and direction of edges and eliminates the prior constraint of the K2 algorithm that requires a given order of nodes, but the cost of the test calculation is exponential. In the second year, he proposed a PC algorithm [13], which improved upon the search strategy of the SGS algorithm. It requires less computational load when learning a sparse network structure, and it was used by Tsagris [14]. Cheng combined the idea of information theory with an independent testing method [15], and the proposed learning method exhibited good performance in structural learning.
- (3) Hybrid algorithm: because the method based on score search and the method based on constraint have their own advantages and disadvantages, the hybrid optimization algorithm combining the two has gradually become the mainstream of research. The improved whale optimization algorithm is used

to optimize the structure of Bayesian network. The optimization efficiency and accuracy of this method are good, but the complexity is very high [16]. The improved particle swarm optimization (PSO) is proposed to learn the Bayesian network structure. After the initial network is constrained by mutual information, the improved PSO algorithm is used to search the optimal Bayesian network, which improves the optimization efficiency. However, due to the instability of the algorithm, the accuracy of the structure cannot be guaranteed [17]. The bird swarm algorithm is used as the search strategy to improve the search strategy, which makes the search ability stronger and the convergence further improved [18].

For the K2 algorithm, as stated by Cooper, the K2 algorithm can reconstruct a moderately complex belief network rapidly, but it is sensitive to the ordering of the nodes [11]. Further information can be found in [19, 20].

This paper presents a new Bayesian network structure learning method based on random node order and genetic incremental search for an optimal path and compares this method with the K2 algorithm. Experiments demonstrate that the method of random node order can yield a better Bayesian network structure without expert knowledge, and the genetic incremental structure learning method can greatly improve the computational efficiency when tested on big datasets, especially when the number of samples and nodes are large. It always exhibits a runtime that is shorter than that of the K2 algorithm.

## 2. K2 Algorithm

The K2 algorithm [21] effectively integrates prior information in the search process and exhibits good time performance. It is a classic structure learning algorithm based on scoring search.

*2.1. Scoring Function.* The node sequence is given in advance, and each node greedily searches its parent node set from its predecessor node according to the Bayesian scoring function and finally obtains the network structure with the best score.

The learning of the network structure can be attributed to the given dataset  $D$  and finds a network structure  $B_s$  with the largest a posteriori probability, that is,  $B_s$  is set to maximise  $P(B_s|D)$ . Additionally,  $P(B_s|D) = P(B_s)P(D|B_s)/P(D)$ , the denominator  $P(D)$  is unrelated to  $B_s$ , so the ultimate objective is to find the maximum  $B_s$  of  $P(B_s|D)$ . Through a series of derivations (for the specific derivation process, see [6]), we obtain

$$P(B_s|D) = P(B_s) \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!, \quad (1)$$

where  $P(B_s)$  is the priori probability of  $B_s$ . This is the probability set for each structure without providing data.  $N_{ijk}$  denotes the  $j$ th unique instantiation of  $\pi_i$  relative to  $D$ .

Suppose there are  $q_i$  such unique instantiations of  $\pi_i$ . Define  $N_{ijk}$  to be the number of cases in  $D$  in which variable  $x_i$  has the value  $v_{ik}$  and  $\pi_i$  is instantiated as  $w_{ij}$ ; the value of  $N_{ij}$  can be obtained using

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}. \quad (2)$$

For equation (1), the first multiplicative symbol traverses each random variable  $x_i$  by  $i$ , and  $n$  denotes the number of random variables.

The second multiplicative symbol traverses all parent variables of  $x_i$  by  $j$ , and  $q_i$  indicates the number of types of  $x_i$  parent variable instances.

The last multiplicative symbol traverses all possible values of the current variable  $x_i$ , where  $r_i$  denotes the number of possible values.

It can be assumed that the probability of each structure obeys a uniform distribution, that is, the probability  $P(B_s)$  is the same constant as  $c$ . Using constant  $c$  to replace  $P(B_s)$ , equation (1) changes to

$$P(B_s|D) = c \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!. \quad (3)$$

The objective is to obtain  $B_s$  that can maximise the posterior probability, as follows:

$$\max_{B_s} [P(B_s, D)] = c \prod_{i=1}^n \max_{\pi_i} \left[ \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \right]. \quad (4)$$

As can be seen from the above equation, as long as the local maximum for each variable is provided, the overall maximum can be obtained. The component of each variable is presented as a new scoring function, as follows:

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!. \quad (5)$$

**2.2. Search Strategy.** The core of Bayesian network structure optimization is to narrow the search scope through search strategy after determining the scoring function. Greedy search algorithm is the most commonly used method. But it is easy to fall into local optimization. In 2017, the authors of [1] proposed adding disturbance factor to local greedy search, and using the idea of genetic algorithm, the metaheuristic method was used to improve the performance of local greedy search. In 2020, the authors of [22] introduced microbial genetic algorithm into Bayesian network structure learning. The undirected graph with most correct edges is calculated by the maximum information coefficient, and it is used as the initial population, and then the excellent individuals in the initial population are retained by using the operator of microbial genetic algorithm. Through the combination of the two, the purpose of learning close to the real network structure from a small amount of data can be achieved.

Search strategy assumes the nodes are ordered. If  $x_i$  precedes  $x_j$ , there can be no edges from  $x_j$  to  $x_i$ . Simultaneously, it assumes that the maximum number of parent variables per variable is  $u$ . Each time the largest parent variable of the scoring function is selected and inserted into the set, the loop is terminated, and the scoring function cannot be increased further.

### 3. Incremental K2 Algorithm Based on Random Attribute Order

**3.1. Random Generate Nodes Order.** The K2 algorithm must initialise the order of nodes such that only the node in front of node  $i$  can be the parent of node  $i$ . This is defined as

$$\text{Pred}(x_i) = x_j (j = 1, 2, \dots, i - 1). \quad (6)$$

The disadvantages are as follows. First, the order of nodes is not easy to obtain in most actual network structures, and the expression of this a priori knowledge is not conducive to the understanding of expert knowledge. Second, the fault tolerance for the order of nodes is poor. If the order of nodes that is input into the K2 algorithm is dissimilar to that in the real structure, the accuracy of the K2 algorithm will be greatly reduced; this is owing to its algorithmic theory. In this study, our first objective is to reduce the dependence on the order of nodes. For nodes, each iteration randomly generates an array of nodes, and the generation procedure (see Algorithm 1) is as follows.

**3.2. Genetic Incremental K2 Algorithm.** The basic idea is to divide the training data into two groups and use the first set of training data to learn a basic Bayesian network structure using the K2 algorithm. In the process of learning, not only the current optimal value, i.e., the decision-making of the algorithm each time, but also several suboptimal values are saved [23], a GA is applied, and the current four optimal values are mutated to a new optimal value. The number of suboptimal values can be adjusted, for example, three or four. This study selects three suboptimal values. When using the second set of incremental data, it will not research; instead, it will take the four optimal values and the new genetic value as the next search space. The algorithm skilfully eliminates the low-level model, reduces the search space, and improves the efficiency of the algorithm.

In addition, the optimal score function value is preserved in each iteration. After the iteration, the node order of the Bayesian network with the optimal score function value is considered to be the best node order.

The algorithm is divided into two parts. The first part (see Algorithm 2) is to generate one optimal value and three suboptimal values, mutate a new optimal value from the first set of data, and store all values in the candidate matrix. The following pseudocode expresses the first part.

The second part (see Algorithm 3) of the algorithm uses the second set of data to optimize the suboptimal value. The core content is as follows.

```

(1) Procedure randomarray(len, start, end) (
(2)   {Input  $n$ ,  $n$  is nodes number.}
(3)   {Output: an array that has  $n$  values between 1 and  $n$ .}
(4)   Define the array[1, $n$ ]: order
(5)   for1 1- $n$  {
(6)     Random generation of a number between 1 and  $n$ 
(7)     If (the number is the first number) {insert the number into the array; }
(8)     Else{
(9)       for2 1-length(array){
(10)        If(the number exists){ regenerate a random number, loop for2}
(11)        else{insert the number into the array; loop for1}
(12)        End for2}
(13)    End else }
(14)  End for1}
(15) End randomarray }

```

ALGORITHM 1: Random array algorithm.

```

(1) Procedure GCDK2{
(2) {Input: K2 algorithm need parameter initialisation, mutation rate  $pm = 0.5$ }
(3) {Output: a bayesian network and a matrix contain optimal value and location }
(4) For  $i = 1-n$ {
(5)    $P_{old} = g(i, \pi_i)$ 
(6)   While OK To Proceed and Length is OK
(7)      $pps = \text{mysetdiff}(\text{order}(1: i - 1), ps)$ {Potential parents}
(8)      $nps = \text{length}(pps)$ 
(9)     For  $pi = 1-nps$ {
(10)       $pscore(pi) = g(i, \pi_i \cup \{z\})$ 
(11)    End for}
(12)    [ $P_{first}$ ,  $best\_p_{first}$ ] =  $\max(pscore)$ ;{ $best\_p_{first}$  is a location,  $P_{first}$  is the max first score}
(13)    Get  $second\_pscore$  and  $second\_p$ 
(14)    Get  $third\_pscore$  and  $third\_p$ 
(15)    Get  $fourth\_pscore$  and  $fourth\_p$ 
(16)    Mutate optimal values get a new genetic value  $GA\_pscore$ 
(17)    If ( $P_{first} > P_{old}$ ) { $P_{old} = P_{first}$ ;  $ps = [ps \text{ best\_}p_{first}]$  }
(18)    Input  $P_{first}$ ,  $best\_p_{first}$ ,  $second\_pscore$ ,  $second\_p$ ,  $third\_pscore$ ,
(19)     $third\_p$ ,  $fourth\_pscore$ ,  $fourth\_p$  and  $GA\_pscore$  into candidate matrix
(20)    End if}
(21)  End while}
(22) End for}
(23) End GCDK2}

```

ALGORITHM 2: GCDK2 algorithm.

#### 4. Experimental Results

To test the algorithm, the general ALARM, Asia, and CANCER networks were selected in the experiment. Under different sample numbers, the running time and structural hamming distance [24] were used to evaluate different algorithms.

The experiment adopted the Bayesian network toolbox in the MATLAB platform. The operating environment was Windows 7, Intel (R) Core(TM) i3-4170, 3.70 GHz CPU, 8.00 GB RAM. The results of the experiment are listed in Tables 1 and 2.

This algorithm relaxes the strict requirement of the K2 algorithm on node order and improves the efficiency of learning the Bayesian network structure.

- (1) In the ALARM network (comprising 37 nodes), the experiment began with a sample size of 4000, the running time of K2 was 7.928, GAK2 was 5.675, and GIMK2 was 0.861; when the size varied to 50000, the times were 29.877, 18.429, and 3.128; when the size was 100000, the times were 55.659, 37.839, and 5.987. SHD is large at first; however, it eventually reduces to zero.

```

(1) Procedure GIMK2(part){
(2)   For  $i = 1$ -length(candidate)
(3)   child = candidate( $i, 1$ )
(4)   For  $j = k$  to  $k+4$  {from  $k$  to  $k+4$  because the candidate stored 5 values}
(5)     ppar = candidae( $i, j$ )
(6)      $u = \text{find}(\text{clamped}(\text{child}, : ) == 0)$ 
(7)     Get  $P_{\text{new}}$  from child and  $u$ 
(8)   End for
(9)   best_action = candidate( $i, \text{best}_p$ )
(10)  End for
(11)  ...
(12) End GIMK2}

```

ALGORITHM 3: GIMK2 algorithm.

TABLE 1: Result of learning ALARM by learn\_struct\_K2 and GIK2.

Sample size	K2		GAK2		GIK2	
	Time (s)	SHD	Time (s)	SHD	Time (s)	SHD
4000	7.928	2	5.675	2	0.861	2
5000	8.348	2	6.253	2	0.831	2
10000	11.047	2	8.754	2	1.110	2
15000	12.740	2	10.546	2	1.348	2
20000	15.456	1	11.328	1	1.547	1
50000	29.877	1	18.429	1	3.128	0
100000	55.659	0	37.839	0	5.987	0

TABLE 2: Result of learning Asia by learn\_struct\_K2 and GIMK2.

Sample size	K2		GAK2		GIK2	
	Time (s)	SHD	Time (s)	SHD	Time (s)	SHD
200	0.234	3	0.205	2	0.058	2
500	0.240	1	0.218	1	0.073	1
1000	0.255	1	0.220	1	0.057	1
3000	0.225	1	0.231	1	0.062	1
5000	0.232	1	0.236	1	0.078	1
10000	0.264	1	0.238	1	0.089	1
15000	0.291	0	0.247	0	0.141	0
20000	0.376	0	0.275	0	0.137	0
50000	0.593	0	0.279	0	0.235	0
100000	1.084	0	0.884	0	0.423	0

- (2) In the Asia network (comprising 8 nodes), the experiment began with a sample size of 200; however, when the size varied to 5000, the running time of K2 was 0.232, GAK2 was 0.236, and GIMK2 was 0.078; when the size varied to 50000, the times were 0.593, 0.279, and 0.235; when the size was 100000, the times were 1.084, 0.884, and 0.423.
- (3) In the CANCER network (comprising 5 nodes), the experiment began with a sample size of 200; however, when the size was 5000, the running time of K2 was 0.158, GAK2 was 0.139 and GIMK2 was 0.053; when the size varied to 50000, the times were 0.248, 0.218, and 0.188; when the size was 100000, the times were 0.353, 0.245, and 0.191.

The results of the program operation indicate that the genetic incremental K2 algorithm has a shorter running time than K2 and GAK2 for the same sample size. When the mobile Bayesian networks with different number of nodes, particularly when the dataset size increases with the number of nodes, the running time of K2 and GAK2 becomes extremely long.

## 5. Conclusion

As data analysis is now being conducted on big data, if we need to analyse big data with uncertain knowledge, especially in the case of numerous attributes, the genetic incremental K2 algorithm can reduce the search space and

considerably improve the efficiency of the algorithm. The improved algorithm in this paper is effective; however, it has disadvantages such as the fact that the search space of each algorithm depends on the current optimal path; thus, it is easy to fall into local optimum. The algorithm should thus combine particle swarm optimization, ant colony, or another optimization algorithm to avoid falling into local optimum.

### Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

This work was supported by the training object of “Blue Project” in Jiangsu Universities in 2019.

### References

- [1] C. Lee and P. Van Beek, “Metaheuristics for score-and-search bayesian network structure learning,” *Advances in Artificial Intelligence*, vol. 10233, no. 5, pp. 129–141, 2017.
- [2] D. Wang, A. AmrilJaharadak, and Y. Xiao, “Dynamic knowledge inference based on bayesian network learning,” *Mathematical Problems in Engineering*, vol. 2020, no. 4, 9 pages, Article ID 6613896, 2020.
- [3] Y. Jiang, X. Gu, D. Wu et al., “A novel negative-transfer-resistant fuzzy clustering model with a shared cross-domain transfer latent space and its application to brain CT image segmentation,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 1, pp. 40–52, 2021.
- [4] L. Li and J. Zhang, “Research and analysis of an enterprise E-commerce marketing system under the big data environment,” *Journal of Organizational and End User Computing*, vol. 33, no. 6, pp. 1–19, 2021.
- [5] X. Yang, H. Li, L. Ni, and T. Li, “Application of artificial intelligence in precision marketing,” *Journal of Organizational and End User Computing*, vol. 33, no. 4, pp. 209–219, 2021.
- [6] Y. Jiang, Y. Zhang, C. Lin, D. Wu, and C.-T. Lin, “EEG-based driver drowsiness estimation using an online multi-view and transfer TSK fuzzy system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1752–1764, 2021.
- [7] D. Wang, X. Yin, and Ya Gao, “Image Denoising method based on NSCT bivariate model and Variational Bayes threshold estimation,” *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8927–8941, 2019.
- [8] H. Qiu, Z. Wei, Y. Liu et al., “A Bayesian network meta-analysis of three different surgical procedures for the treatment of humeral shaft fractures,” *Medicine*, vol. 95, no. 51, Article ID e5464, 2016.
- [9] J. Xu, Z. Li, and An Xu, “KDE-CGA algorithm of structure learning for small sample data bayesian network,” *Computer Science*, vol. 44, no. 11A, pp. 437–441, 2017.
- [10] B. Liu, H. Wang, M. Sun, H. Liu, Y. Liu, and C. Zhang, “Learning bayesian network structure from node ordering searching optimal,” *Journal of Electronics and Information Technology*, vol. 40, no. 5, pp. 1234–1241, 2018.
- [11] G. F. Cooper and E. Herskovits, “A Bayesian method for the induction of probabilistic networks from data,” *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.
- [12] P. Spirtes, C. N. Glymour, and R. Scheines, “Causality from probability,” *Carnegie-Mellon University, Laboratory for Computational Linguistics*, vol. 112, 1989.
- [13] P. Spirtes and C. Glymour, “An algorithm for fast recovery of sparse causal graphs,” *Social Science Computer Review*, vol. 9, no. 1, pp. 62–72, 1991.
- [14] M. Tsagris, “Bayesian network learning with the PC algorithm: an improved and correct variation,” *Applied Artificial Intelligence*, vol. 33, no. 2, pp. 101–123, 2019.
- [15] J. Cheng, R. Greiner, J. Kelly, D. Bell, and W. Liu, “Learning Bayesian networks from data: an information-theory based approach,” *Artificial Intelligence*, vol. 137, no. 1-2, pp. 43–90, 2002.
- [16] W. D. Wang, N. Ai-Dhahir, H. Minn, and N. Al-Dhahir, “A game-theoretic approach for exploiting multiuser diversity in cooperative slotted aloha,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4215–4225, 2018.
- [17] Y. F. Liu, H. Jin, Q. M. Yu, and W. Zheng, “Research on transmission waveform structure and rate scaled of new generation data link,” in *Proceedings of the 2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1686–1689, IEEE, Changsha, China, July 2017.
- [18] H. Chen and Na Zhang, “Bayesian network structure learning based on hybrid improved bird swarm algorithm,” *Journal of Air Force Engineering University (Natural Science Edition)*, vol. 22, no. 1, pp. 85–98, 2021.
- [19] Y. Liu, Z. Li, B. Wei, X. Li, and B. Fu, “Seismic vulnerability assessment at urban scale using data mining and GIScience technology: application to Urumqi (China),” *Geomatics, Natural Hazards and Risk*, vol. 10, no. 1, pp. 958–985, 2019.
- [20] L. Lv and Q. Lu, “Rapid method to build bayesian network structure based on attribute order relation,” *Comput. Eng. Des.*, vol. 39, pp. 2961–2966, 2018.
- [21] D. Shan, Q. Lv, and F. Li, “An effective mountain climbing algorithms in bayesian network learning,” *Minicomput. Syst.*, vol. 12, pp. 32–33, 2009.
- [22] M. Wu and Ji Liu, “Bayesian network structure learning based on microbial genetic algorithm,” *Journal of Chongqing Institute of science and technology (Natural Science Edition)*, vol. 22, no. 6, pp. 70–74, 2020.
- [23] A. Yasin and P. Leray, “iMMP: a local search approach for incremental bayesian network structure learning,” *Advances in Intelligent Data Analysis X in Proceedings of the 10th international conference on Advances in intelligent data analysis X*, vol. 10, pp. 401–412, Springer Berlin Heidelberg, Porto, Portugal, October 2011.
- [24] C. Barrett, Q. He, F. W. Huang, and C. M. Reidys, “An efficient dual sampling algorithm with hamming distance filtration,” *Journal of Computational Biology*, vol. 25, no. 11, pp. 1179–1192, 2018.