

On the Hyper-Parameters and the Scoring Functions of PCTABU and PCHC Bayesian Network Algorithms

Volkan Sevinç

vsevinc@mu.edu.tr

Muğla Sıtkı Koçman University, Faculty of Science

Michail Tsagris

University of Crete, School of Social Sciences

Research Article

Keywords: Causality, Bayesian networks, PCTABU, PCHC, scoring functions

Posted Date: October 26th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11352321/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

On the Hyper-Parameters and the Scoring Functions of PCTABU and PCHC Bayesian Network Algorithms

Volkan Sevinç¹ Michail Tsagris²

¹*Muğla Sıtkı Koçman University, Faculty of Science, Department of Statistics, Kötekli Kampüsü, 48000, Muğla, Türkiye, vsevinc@mu.edu.tr*

²*University of Crete, Department of Economics, School of Social Sciences, 74100, Rethymno, Crete, Greece, mtsagris@uoc.gr*

ABSTRACT

The goal of this paper is to introduce a novel Bayesian network learning algorithm, and to examine the effects of scoring function on Bayesian networks by comparing the estimation performance of the newly introduced algorithm and a previously suggested algorithm. The comparison was performed using both simulated and real-world-data-based Bayesian networks. The results show that the scoring function can have a significant effect on the performance of Bayesian networks. The newly introduced algorithm appears to perform better for networks with lower average Markov blanket sizes, average degrees, and densities.

Keywords: Causality, Bayesian networks, PCTABU, PCHC, scoring functions

1. Introduction

Causal relationships among variables are central to empirical research across various disciplines, particularly in economics (Hoover, 2008; Heckman, 2008). Bayesian networks (BNs) have

emerged as a prominent framework for uncovering such relationships, representing variables as nodes in a directed acyclic graph (DAG) whose edges encode conditional dependencies. Their ability to combine expert knowledge with data-driven inference has made them a widely adopted tool across diverse applied domains, including finance, marketing, and macroeconomics (Gemela, 2001; Cugnata et al., 2014; Teye and Ahelegbey, 2017).

A substantial body of work has been devoted to BN learning, that is, the problem of inferring the DAG from data. Existing algorithms fall into three broad categories. Constraint-based methods, such as the PC algorithm (Spirtes and Glymour, 1991) and interleaved incremental association Markov blanket (interleaved IAMB) (Yaramakala and Margaritis, 2005), build the network skeleton through sequential conditional independence (CI) tests. Score-based methods, such as those relying on the K2 score (Cooper and Herskovits, 1992), Bayesian Dirichlet equivalent (BDe) (Heckerman et al., 1995), log-likelihood (LL) (Friedman et al., 1997), or Bayesian information criterion (BIC) (Schwarz, 1978), evaluate candidate structures by optimizing a global scoring function using search algorithms such as hill-climbing (HC) or Tabu search (TABU). Hybrid methods combine both approaches: they use CI tests to construct an initial skeleton, and then apply score-based optimization to orient edges and refine the structure. Notable hybrid algorithms include the max-min HC (MMHC) (Tsamardinos et al., 2006) and PCHC (Tsagris, 2021), the latter of which integrates the PC skeleton construction procedure with HC-based score optimization and offers computational advantages over its predecessors.

Despite this progress, two important questions have received comparatively little systematic attention. First, the choice of optimization algorithm used in the score-based phase, HC or TABU, may substantially affect how the network search space is explored, yet its impact

on structure learning performance has not been rigorously evaluated in a hybrid setting. Second, the choice of scoring function (BDe, LL, or BIC) introduces further variability whose practical consequences are not well understood. Both choices operate as hyper-parameters in the sense that they are user-selected prior to learning and can meaningfully alter the learned network, yet neither is routinely treated as an object of systematic study.

This paper addresses both gaps. We introduce PCTABU, a novel hybrid algorithm that integrates the skeleton construction stage of the PC algorithm with TABU-based score optimization, and compare it systematically with PCHC across three scoring functions, the BDe, LL, and BIC, using both simulated datasets and real-world benchmark networks. The results show that hyper-parameter and scoring function choices have significant effects on BN structure learning performance. PCTABU tends to outperform PCHC for networks with lower average Markov blanket sizes, average degrees, and densities, while the LL scoring function consistently produces the worst results. BDe and BIC perform similarly across most settings. The remainder of the paper is organized as follows. Section 2 presents the relevant methodological background and describes the PCTABU algorithm. Section 3 discusses the experimental design, and the findings of the comparison, while Section 4 concludes the paper.

2. Materials and Methods

In this section, information about the theoretical background of BNs, the optimization algorithms, and the scoring functions are provided.

2.1. BNs

BNs, introduced by Pearl (1988), are a class of graphical models represented as DAGs. In DAGs, edges are used to indicate the relationship between two variables (or vertices) as $V_i - V_j$. In a DAG these relationships are indicated by arrows, which also represent the directed relations as $V_i \rightarrow V_j$ on the condition that these arrows do not constitute a cyclic pattern.

BNs consist of two main components: a network structure and a probabilistic component. Let $B = (G, \mathbf{P})$ denote a BN, where G denotes a DAG over K nodes (vertices or variables) $\mathbf{V}$, which have a joint probability distribution (JPD) P , that defines the BN and can be expressed as follows.

$$P(V_1, V_2, \dots, V_K) = \prod_{i=1}^K P(V_i | Pa(V_i)) \quad (1)$$

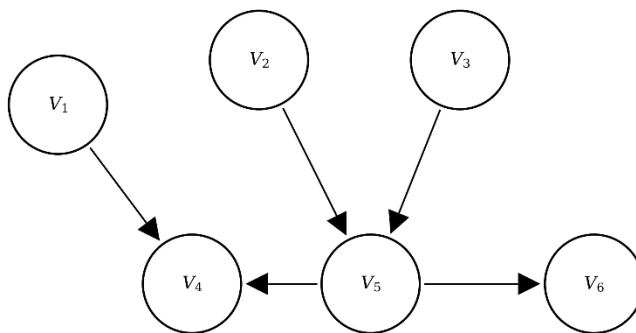
where V_i denotes the i -th node in the network, K denotes the number of nodes in the network, and $Pa(V_i)$ denotes the set of parents of node V_i . One of the fundamental assumptions in BNs is faithfulness. If G entails only CIs in P and all CIs in P are entailed by G , based on the Markov condition, then G and P are faithful to each other, and G is a perfect map of P . When these two conditions are not satisfied, then P and G are considered to be unfaithful to each other (Spirtes et al., 2000, Neapolitan, 2004).

In discrete BNs, the JPD is displayed in tabular forms called conditional probability tables (CPTs). In contrast, continuous BNs represent probabilistic relationships using parameterized probability density functions instead of finite tables. With continuous variables, the conditional distributions are typically specified within families such as linear Gaussian models, where each node is associated with a conditional distribution whose parameters depend on its parent variables. This functional representation replaces enumerative tabulation with

analytically defined distributions, thereby enabling inference over an uncountably infinite state space.

For mixed BNs, which incorporate both discrete and continuous variables within the same graphical structure, the representation becomes correspondingly more intricate. In such models, discrete variables often govern the selection of conditional density regimes for continuous descendants, resulting in mixtures of parametric distributions conditioned on discrete parent configurations. Conversely, continuous parents may influence discrete children through conditional probability models defined by link functions or threshold mechanisms. The joint distribution in mixed settings is therefore constructed from a coherent combination of conditional probability mass functions and conditional probability density functions, ensuring consistency with the directed acyclic graph structure while preserving measure-theoretic validity across mixed domains.

The structure of the example BN is represented as a DAG, as illustrated in Figure 1, where nodes correspond to random variables and directed edges encode conditional dependencies consistent with the acyclicity property.



100 The nodes from which the arrows are directed to V_i are called parents. Consequently, the
 101 node V_i is called child node of its parent nodes. The BN in Figure 1 has six nodes $V_1, \dots, V_6$ and,
 102 for example, nodes V_2 and V_3 are the parents of V_5 . On the other hand, V_4 is the child node of V_1
 103 and V_5 . The other parents of a node's child are called its spouse nodes; in Figure 1, V_2 and V_3 are
 104 spouses of each other, since both are parents of V_5 .

105 In causal BNs, edges are interpreted causally. That is, an edge $V_i \rightarrow V_j$ exists if V_i is a
 106 direct cause of V_j . The triplet (V_i, V_k, V_j) where $V_i \rightarrow V_k \leftarrow V_j$, on the other hand, is called V-
 107 structure and if there is no edge between V_i and V_j , the node V_k is called unshielded collider. This
 108 implies that V_i and V_j are dependent given V_k , whereas they are independent given any separating
 109 set that contains neither it nor its descendants, provided that G and P satisfy the faithfulness
 110 condition (Spirtes et al., 2000). In Figure 1, $V_2 \rightarrow V_5 \leftarrow V_3$ is an example of V-structure and V_5 is
 111 an unshielded collider. The triplet of nodes (V_i, V_k, V_j) such that $V_k \rightarrow V_i$ and $V_k \rightarrow V_j$ is called Λ -
 112 structure. The Λ -structure implies that V_i and V_j are conditionally independent given V_k . One
 113 such example in Figure 1 is the triplet (V_4, V_5, V_6) where $V_5 \rightarrow V_4$ and $V_5 \rightarrow V_6$.

114 Multiple BNs may encode the same set of CIs. Such BNs are called Markov equivalent.
 115 The set of all Markov equivalent BNs forms the Markov equivalence class, which can be
 116 represented by a completed partially directed acyclic graph (CPDAG). Markov equivalent DAGs
 117 share the same skeleton and the same V-structures. In addition to directed edges, the CPDAG
 118 also contains undirected edges. An undirected edge indicates that the edge may be oriented in
 119 either direction among the BNs in the Markov equivalence class (provided that no new V-

structure or directed cycle is created), whereas the directed edges and the missing edges are shared among all equivalent BNs.

2.2. BN learning

In hybrid approaches, BN learning involves two main phases. In the first phase, the skeleton of the network is constructed, or identified, through various algorithms, where potential edges between nodes are identified using CI tests. In the second phase, the objective is to derive the optimal DAG structure by applying scoring functions. Edges are either converted into directed arrows or removed through the optimization of these scoring functions. The optimization process is guided by certain algorithms which maximize the scoring functions.

Among the early and influential contributions to BN structure learning, Cooper and Herskovits (1992) introduced the K2 algorithm, one of the first score-based approaches for learning BN structures from data. The K2 algorithm employs a scoring metric and a greedy search (GS) procedure to identify parent sets for each node under a predefined variable ordering. This work established a foundation for subsequent score-based BN learning methods. Later, Chickering (1996, 2002) made significant contributions to the field by investigating the theoretical properties of BN structure learning and developing efficient search strategies in the space of equivalence classes of DAGs. In particular, Chickering (1996, 2002) demonstrated the complexity of optimal BN structure learning and proposed algorithms that improved the efficiency and accuracy of score-based search procedures. These studies have become fundamental references in the BN learning literature and have substantially influenced the development of modern hybrid and score-based algorithms.

In BN learning, the choice of algorithm, particularly those used for score optimization, plays a critical role in determining the final network configuration. While such algorithms are procedural optimization methods rather than tunable numerical hyper-parameters, their selection can substantially influence how the network space is explored and how dependencies among variables are identified. Different optimization algorithms navigate the combinatorial space of possible networks in distinct ways, potentially leading to variations in both the qualitative structure of the network and its quantitative fit to the data. Consequently, the choice of an optimization algorithm can be regarded similarly to a hyper-parameter: it represents a user-selectable option that affects the learning outcome and can be adjusted according to dataset characteristics, computational constraints, or modeling objectives. By explicitly acknowledging this perspective, the methodology provides a more nuanced understanding of the network construction process and clarifies how algorithmic decisions impact model behavior. The following subsection presents a detailed overview of two relevant hyper-parameters, namely HC and TABU.

2.3. HC and TABU Algorithms

HC and TABU are two types of optimization algorithms that can be used to maximize the scoring functions of a BN, after constructing its skeleton.

The HC phase performs a GS over the space of DAGs¹. The process assigns directions to the edges, or even deletes edges, but if an operation creates a cycle, the operation is cancelled. The assessment score (e.g. BIC, LL, BDe, or another score) is computed and the edge operation is accepted if the score increases, continuing until the score cannot further increase.

¹ For a detailed discussion of the HC algorithm the reader is addressed to (Adhitama & Saputro, 2022).

GS can be extended by TABU search (Glover, 1986), which allows the search to escape local maxima. The TABU algorithm starts from a feasible initial solution and selects a series of specific operations (edge orientations) moving in different directions for an exploratory search. If an operation in a certain direction results in the highest score increase, that solution is locally optimal. That solution is then entered into the TABU list (Friedman, Nachman and Pe'er, 1999), and the initial solution is replaced with the new optimal solution. This process is repeated and the TABU list is updated until the convergence criterion is met. The list keeps the last 100 structures explored (in the experiments of Section 3, the list size was set to 10). Instead of applying the best local change, the best local change that results in a structure not on the list is performed to escape local maxima. This change may reduce the score. When 15 changes occur (10 in the experiments of Section 3) without an increase in the maximum score ever encountered during search, the algorithm terminates. The overall best scoring structure is then returned.

2.4. The PCHC algorithm

The skeleton identification phase in the PCHC algorithm mirrors that of the original PC algorithm, and its procedure is outlined in Algorithm 1. This phase begins by assessing all pairwise unconditional dependencies among variables. Edges are removed between variable pairs that do not exhibit statistically significant associations. Following this initial pruning, the algorithm proceeds with CI tests, where the size of the conditioning set (denoted as k) is incrementally increased by one at each iteration. During each iteration, conditioning sets are formed from all possible subsets of the adjacent variables denoted as $adj(G, V_i)$, where V_i is the variable under consideration.

183 Importantly, the algorithm operates on an ordered set of variable pairs, ensuring
184 deterministic selection and evaluation. This iterative process continues until no further edges can
185 be removed from the graph.

186 As noted by Glymour (2001), the method used to select and order variable pairs is
187 critical, as improper heuristics may lead to incorrect causal structures. Among the three primary
188 heuristics proposed, the most effective involves testing, for a given variable V_i , those variables V
189 that exhibit the weakest probabilistic dependence on V_i , conditioned on subsets that demonstrate
190 the strongest dependence. This approach ensures a deterministic and consistent identification of
191 the network skeleton, making the results independent of the initial ordering of variables in the
192 dataset (Tsagris, 2019).

193 The PC algorithm is a constraint-based algorithm introduced by Spirtes and Glymour
194 (1991). Following the logic of constraint-based algorithms, the PC algorithm initially places all
195 edges among the variables in the data set. It then applies CI tests among the nodes to remove
196 edges between node pairs that exhibit no significant dependency. To test independence with
197 discrete data, PC algorithm primarily uses the G^2 test of independence.

198 MMHC algorithm, on the other hand, initially forms the skeletons of all possible
199 networks by applying the independence tests, using the max-min parents and children (MMPC)
200 constraint-based algorithm. However, in the second phase, it calculates a score for each
201 candidate network by using the HC search, and it chooses the network having the highest score.
202 However, the MMPC algorithm can impose a substantial computational burden when the
203 network contains many nodes.

204 To mitigate this burden, Tsagris (2021) proposed a new scalable BN algorithm named
 205 PCHC, which has the same mechanism to build the skeleton form of the BN as PC algorithm.
 206 However, afterwards, it uses HC to perform selection among the candidate networks by
 207 maximizing the scoring function. The notable difference between PCHC and MMHC is that the
 208 former uses the PC algorithm, whereas the latter uses the MMPC algorithm, during the skeleton
 209 identification phase.

210 **Algorithm 1.** Skeleton identification phase of the PC algorithm.

-
- 1: **Input:** Data set on a set of K variables $\mathbf{V}$.
 - 2: Let $k = 0$ and assume all variables are connected to each other.
 - 3: **Repeat**
 - 4: Select an ordered pair of variables V_i and V_j such that
 - 5: $|adj(G, V_i) \setminus \{V_j\}| \geq k$, and a subset $\mathbf{S}$ with $|\mathbf{S}| = k$
 - 6: If $(V_i \perp\!\!\!\perp V_j | \mathbf{S})$ delete edge $V_i - V_j$ from G .
 - 7: Until all ordered pairs of adjacent variables V_i and V_j with $|adj(G, V_i) \setminus \{V_j\}| \geq k$ and
 all subsets $\mathbf{S}$ with $|\mathbf{S}| = k$ have been tested for conditional
 independence.
 - 8: $k=k+1$
 - 9: **Until** for each ordered pair of adjacent variables V_i and V_j , $|adj(G, V_i) \setminus \{V_j\}| < k$.
 - 10: Return G , sepset.
-

211 2.5. The PCTABU algorithm

The proposed PCTABU algorithm², on the other hand, uses the skeleton-construction mechanism of the PC algorithm, but employs the TABU algorithm as the hyper-parameter at the DAG selection phase. The skeleton identification phase of PCTABU, which follows the skeleton identification procedure of the PC algorithm (Spirtes and Glymour, 1991), is provided in Algorithm 1. Building on the same idea of incorporating a scoring function into the PC algorithm framework, the PCTABU algorithm uses the TABU algorithm to select the final network by maximizing a score assigned to each candidate BN skeleton.

The final network structure is determined through the application of various optimization algorithms. The behavior of these optimization algorithms is governed by hyper-parameters, such as the tabu list size and the maximum number of iterations without improvement. These hyper-parameters are not directly inferred from the data; they control the search process and dictate how the algorithm explores the solution space in order to identify an optimal or near-optimal solution.

2.6. BN scoring functions

BN learning algorithms are classified into three categories: constraint-based, score-based, and hybrid methods. In general, the constraint-based algorithms form the structure and detect the direction of the edges of the BN by applying various CI tests among the nodes. Some examples of constraint-based algorithms are PC (Spirtes and Glymour, 1991), fast causal inference (FCI) (Spirtes et al. 2000), Grow-Shrink (Margaritis, 2003), and the interleaved IAMB (Yaramakala and Margaritis, 2005). The score-based algorithms start with forming various candidate network

² Both the PCHC and PCTABU algorithms are available in the *R* package *pchc* (Tsagris (2026)).

structures. They assign a score to each candidate network structure using a scoring function and search the space of structures (e.g., by HC or TABU) for the one with the highest score, which is adopted as the final output network.

Some examples of the suggested scoring functions in the literature are K2 score (Cooper and Herskovits, 1992), BDe score (Heckerman et al., 1995), LL (Friedman et al., 1997), and BIC (Schwarz, 1978). Hybrid algorithms, on the other hand, operate as a combination of the constraint-based and score-based algorithms by applying CI tests to form the structure of the network in the first phase, then, in the second phase, they assign scores to decide for the final network having the highest score. Some examples of hybrid-methods are given by Singh and Valtorta (1995), Dash and Druzdzel (1999), Acid and De Campos (2001), Tsamardinos et al. (2006) and Tsagris (2021).

2.6.1. The BDe scoring function

While developing the BDe scoring function, Heckerman et al. (1995) relied on two assumptions: likelihood equivalence and structure possibility. The term equivalence can be explained as follows.

Definition 1 (equivalence): Two directed acyclic graphs are said to be equivalent if they can encode the same joint probability distributions. Let G denote a BN, and the data can be viewed as a multinomial sample of the joint space $\mathbf{D}$ with parameters

$$\Theta_{\mathbf{D}} = \{\theta_{V_1, \dots, V_K}\} \quad V_i = 1, \dots, r_i, i \in 1, \dots, K \quad (2)$$

252 where

$$253 \quad \theta_{V_1, \dots, V_K} = \prod_{i=1}^K \theta_{V_i | Pa(V_i)}. \quad (3)$$

254 **Assumption 1** (likelihood equivalence): Given two directed acyclic graphs G and G' , such
 255 that $P(G) > 0$ and $P(G') > 0$, if G and G' are equivalent, then $\rho(\Theta_{\mathbf{D}}|G) = \rho(\Theta_{\mathbf{D}}|G')$.

256 **Definition 2:** The skeleton of a DAG is defined as the undirected graph resulting from ignoring
 257 the directionality of every edge and a DAG is said to be complete if its skeleton is complete.

258 **Assumption 2:** For any complete DAG G , we have $P(G) > 0$.

259 **Theorem 1** (BDe Score): Suppose that $\rho(\Theta_{\mathbf{D}}|G)$ is a Dirichlet with equivalent sample size N' for
 260 some complete directed acyclic graph G in $\mathbf{D}$. Then for any BN G in $\mathbf{D}$, BDe score is calculated
 261 as follows (Heckerman et al., 1995).

$$262 \quad BDe(G) = P(G) \times \prod_{i=1}^n \prod_{j=1}^{q_i} \left(\frac{\Gamma(N'_{ij})}{\Gamma(N_{ij} + N'_{ij})} \times \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + N'_{ijk})}{\Gamma(N'_{ijk})} \right), \quad (4)$$

263 where

$$264 \quad N'_{ijk} = N' \times P(V_i = V_{ik}, Pa(V_i)), \quad (5)$$

265 where N' , which is the equivalent sample size, indicates the strength of our belief in the prior
 266 distribution. In Eq. (5), the probability is evaluated under the prior distribution, with the i -th
 267 variable in its k -th state and its parents in their j -th configuration. The numbers of parent
 268 configurations and states and the counts in Eq. (4) are defined below, in connection with Eqs. (6)
 269 and (7), and the prime denotes the corresponding prior counts.

270 2.6.2. LL scoring function

271 The LL is a statistical measure derived from the likelihood function and is widely used for
272 parameter estimation and model evaluation. The LL basically measures the degree of model fit.
273 In BNs, the LL of the data given a candidate BN structure is one component of the minimum
274 description length (MDL), which is the sum of the length necessary to represent the network and
275 the length required to represent the data given the network. That is, to represent a BN, a joint
276 probability function is used, whose probabilities are stored in CPT tables. This storing process
277 requires a length, which is proportional to the number of free parameters of the factorized JPD.
278 This number is called network complexity (De Campos and Friedman, 2006). The network
279 complexity $C(G)$ can be represented as follows.

$$280 \quad C(G) = \sum_{i=1}^n (r_i - 1)q_i, \quad (6)$$

281 where n denotes the number of nodes in the network, q_i is the number of possible configurations
282 of the parents of the node, and r_i is the number of levels of the node V_i .

283 The generally used proportionality factor (Rissanen, 1986) is $\frac{1}{2}\log(N)$ and then the
284 description length of a network can be defined as $\frac{1}{2}C(G)\log(N)$.

285 Thus, the length required to represent the data given the network is the negative LL of the
286 data with respect to the network, and the MDL is the negative LL plus the network description
287 length above (which leads to the BIC score of Section 2.5.3). Naturally, the negative LL is
288 expected to be minimal for a fixed network that has been estimated from the data set itself by
289 using maximum likelihood. The LL scoring function is given by

$$LL(G) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \left(\frac{N_{ijk}}{N_{ij}} \right), \quad (7)$$

where N_{ijk} is the number of cases in the data in which the i -th variable is in its k -th state and its parents are in their j -th configuration, and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$ is the marginal count for the parent configuration.

The LL scoring function is less effective in learning the structure of the network, as it tends to favor complete graph structures, in which every node is connected to every other node. This situation, however, is quite undesirable, as these networks do not represent the independence relations in the learned distributions. In addition, such distributions contain a large number of parameters. The parameters that are learned in a complete network match the training data perfectly; however, they do not fit the test data well, causing a problem called overfitting (Friedman et al., 1997).

2.6.3. The BIC scoring function

The BIC scoring function is another score, which is based on Schwarz information criterion (Schwarz, 1978). The main idea of the BIC score is to find the network structure that best fits the data penalized by the number of parameters required for specifying the joint probability distribution given by Eq. (1). This idea can be extended into a scoring function $g(G)$ given below.

$$g(G) = LL(G) - f(N)|G|, \quad (8)$$

where $f(N)$ is a non-negative penalizing function, and $|G|$ is the number of free parameters in the network, given by $\sum_{i=1}^n (r_i - 1)q_i$, where r_i is the number of states of node V_i and q_i is the

310 number of parent configurations of V_i . If $f(N) = 1/2 \log(N)$ in Eq. (8), $g(G)$ becomes the BIC
311 scoring function

$$312 \quad BIC(G) = LL(G) - 1/2 \log(N)|G|. \quad (9)$$

313 **3. Comparing the performances of PCTABU and PCHC algorithms**

314 In this section, information about building the novel algorithm PCTABU is provided first. The
315 comparison results between the estimation performances of PCTABU and PCHC algorithms
316 under BDe, LL, and BIC scoring functions are then presented in terms of the performance
317 metrics.

318 PCTABU and PCHC algorithms use two different types of hyper-parameters, TABU and
319 HC, respectively. Therefore, to see the effects of the hyper-parameter and scoring function
320 choices on the performances of the algorithms, two comparison studies based on real-world-data-
321 based and simulated-data-based BNs, created by PCTABU and PCHC with three scoring
322 functions BDe, LL, and BIC, were performed. Both the HC and TABU algorithms used 10
323 restarts, and there was no constraint on the maximum number of iterations allowed until
324 convergence. The TABU list size was set to 10. In addition, the maximum number of iterations
325 that TABU could perform without improving the best network score was also set to 10. The
326 performance metrics used were the SHD and its three components, number of additions,
327 deletions and sign reversals, the F-score and its two components, precision and recall, and the
328 computational cost (running time) measured in seconds.

329

3.1. Performance comparison using real-world-data-based BNs

Table 1 presents the eight real BN, used to simulate data, and PCHC and PCTABU algorithms were used to learn the DAGs. Categorical data sets were generated from these BNs at various sample sizes (10,000, 20,000, 50,000, 100,000, 200,000, 500,000, 1,000,000, 2,000,000, and 5,000,000).

The algorithms were coupled with the three aforementioned scoring functions, i.e. BDe, LL, and BIC; and the accuracy of the estimated DAGs was assessed using the F-score and the SHD metrics and their components, precision, recall for F-score and number of additions, deletions and reversals for the SHD. The computational cost (running time in seconds is also reported). The results are aggregated over 200 replications. The *R* packages *bnlearn* (Scutari and Denis J.B., 2021), *pcalg* (Kalisch, 2012), *pchc* (Tsagris, 2026), and *Rfast2* (Papadakis et al., 2026) were used.

There are two reasons for choosing large sample sizes. The starting sample size was set to 10,000 because the networks entail too many nodes and parameters and small sample sizes would increase the probability of zero frequencies in many combinations. The maximum sample size was set to 5 million because interest lies in the case of large-scale data and how accurately can one learn the true BN from observed data. In modern times, where large scale and big data are frequently met, the computational cost and scalability of an algorithm have become nearly equally important as its accuracy. Assessing an algorithm only in terms of accuracy, with small sample sizes, is no longer indicative of the algorithm's capabilities.

Table 1. Information on the real-world-data-based BNs³ and performance comparison of the hyper-parameters of the two algorithms.

BN	Number of nodes	Number of arcs	Average Markov blanket	Average degree	Better Performance: PCTABU or PCHC
Alarm	37	46	3.51	2.49	PCTABU
Andes	223	338	5.61	3.03	TIE
Child	20	25	3.00	2.50	PCTABU
Hailfinder	56	66	3.54	2.56	PCTABU
Hepar2	70	123	4.51	3.51	TIE
Insurance	27	52	5.19	3.85	TIE
Pigs	441	592	3.66	2.68	TIE
Win95pts	76	112	5.92	2.95	TIE

As shown in Table 1, in the five networks based on the real-world data (Andes, Hepar2, Insurance, Pigs and Win95pts), both algorithms performed almost equivalently, indicating no significant differences, whereas PCTABU outperformed PCHC in the other three networks (Alarm, Child, and Hailfinder). It is worth noting that in all cases there was a strong agreement between the F-score and SHD metrics. Perhaps a feature explaining why PCTABU performed better is that in those cases, the networks had small average Markov blanket and average degrees, regardless of the number of nodes. Therefore, one could argue that in less dense networks PCTABU performs better than PCHC while in denser networks they perform equally well. With regards to their time complexity (results not presented), the running times required are similar for both algorithms. Tables A1-A8 (Appendix A) show that, in general, the BDe and BIC scores perform nearly equally well, but the LL scoring function performs slightly worse than the BDe and the BIC, and this is true for both PCHC and PCTABU algorithms.

³ The BNs were downloaded from the [bnlearn](https://bnlearn.org/) repository.

3.2. Performance comparison using simulated-data-based BNs

To make the simulation-based comparisons, categorical data sets were generated from BNs at various sample sizes (10,000, 20,000, 50,000, 100,000, 200,000, 500,000, 1,000,000, and 5,000,000). Each sample size was coupled with 30, 50, and 70 nodes; as well as 3 and 5 average number of neighbors. For each network and sample size, both BN learning algorithms were estimated; afterwards, the estimated networks were calibrated against the true networks. The same performance metrics (F-score, SHD and their components, including computational cost) were computed and averaged over 200 repetitions.

3.3. Results and discussion

Table 2 summarizes the comparison results of the two algorithms resulting from the simulated categorical data sets (Full information on the results is provided in Tables B1-B6 in the Appendix B).

Table 2. Information on the simulated data-based BNs experimented and performance comparison of the hyper-parameters of the two algorithms.

Number of nodes (p)	Average number of neighbors	Better Performance PCTABU or PCHC
30	3	TIE
30	5	TIE
50	3	TIE
50	5	TIE
70	3	TIE
70	5	TIE

3.3.1. Real-world-data-based BNs

The comparison across eight benchmark networks reveals a clear pattern: PCTABU and PCHC perform equivalently in denser networks (Andes, Hepar2, Insurance, Pigs, and Win95pts), while PCTABU outperforms PCHC in the three sparser networks (Alarm, Child, and Hailfinder). The networks where PCTABU holds an advantage are precisely those characterized by lower average Markov blanket sizes and lower average degrees, regardless of the total number of nodes. This finding is consistent across both the F-score and SHD metrics, which is reassuring given that the two measures capture complementary aspects of structural accuracy.

A plausible explanation for this pattern lies in the way TABU and HC navigate the scoring landscape during the optimization phase. In sparse networks, the PC-derived skeleton constrains a relatively small but non-trivial set of candidate DAG structures, and the optimization phase must choose among a limited number of possible edge orientations. HC, as a greedy procedure, accepts only moves that yield an immediate improvement in the scoring function and terminates as soon as no such move exists. In regions of the scoring landscape that are flat or contain saddle points (these are more likely to arise in sparse networks where individual edges carry more structural weight) the HC risks early convergence at a local optimum. TABU, by maintaining a list of recently visited structures and forbidding their revisitation, continues to explore the neighborhood even in the absence of immediate improvement. This mechanism allows TABU to escape local optima that would trap HC, and in a sparse network where the global optimum may be separated from a local optimum by only a small number of edge reversals, this exploratory advantage translates into meaningfully better structural recovery. In denser networks, by contrast, the search space grows combinatorially and the quality of the PC-derived skeleton becomes the dominant determinant of the final structure. Under these conditions, both algorithms are constrained by the same skeleton and the marginal advantage of

TABU's broader search is absorbed by the shared structural limitations, resulting in comparable performance.

Regarding the scoring functions, the results from the real-world networks confirm that BDe and BIC perform nearly equivalently across all eight benchmarks and for both algorithms, while LL consistently performs worse. This inferiority of LL is not accidental but reflects a fundamental difference in how the three criteria balance model fit against structural complexity. LL measures only the goodness of fit of the network to the observed data, with no penalty for the number of parameters introduced by additional edges. As a result, LL systematically favors denser structures, accepting edges whenever they produce even a marginal improvement in likelihood regardless of whether those edges reflect genuine conditional dependencies or spurious associations. This tendency toward overfitting manifests directly in the structural metrics: LL-based networks accumulate false positive edges inflating SHD and reducing precision. BIC avoids this by subtracting a penalty proportional to the number of free parameters scaled by the logarithm of the sample size, explicitly discouraging unnecessary complexity. BDe imposes a similar regularizing effect through its Bayesian prior, which assigns lower prior probability to denser graphs. Both penalties act as implicit Occam's razor, guiding the optimization toward sparser structures that more faithfully represent the true CI structure. The practical implication is straightforward: practitioners should prefer BDe or BIC over LL when applying PCHC or PCTABU to real categorical data.

3.3.2. Simulated-data-based BNs

The simulation study, conducted across four combinations of network size ($p = 50$, and 70 nodes) and average neighborhood size of 3 and 5 (3 and 5 neighbors on average), yields a

broadly consistent picture: PCTABU and PCHC perform on a par across all simulated configurations, with neither algorithm achieving a clear advantage in any of the six settings examined (Table 2). This result initially appears to contrast with the real-world findings, where PCTABU outperformed PCHC in the three sparser networks. The apparent discrepancy is, however, interpretable within the same framework. The simulated networks, even at the lower neighborhood size of 3, are synthetically generated random graphs whose structural properties (the distribution of Markov blanket sizes and local connectivity patterns) may not replicate the particular sparsity characteristics of Alarm, Child, and Hailfinder that gave TABU its advantage. In other words, the simulated sparse settings (3 neighbors on average) may still be sufficiently regular and symmetric that the scoring landscape does not produce the local optima traps that TABU is better equipped to escape. The real-world networks, by contrast, exhibit more heterogeneous local structures, and it is in those irregular regions that TABU's neighborhood avoidance mechanism is most valuable.

The results for the scoring functions under the simulated settings reinforce the conclusions drawn from the real-world analysis. Across all six configurations and for both algorithms, LL produces worse structural recovery than BDe and BIC, which again perform similarly to each other. This corroborates the interpretation that LL's lack of a complexity penalty leads to systematic overfitting, an effect that is reproducible under controlled simulation conditions and is therefore not an artifact of the particular characteristics of the benchmark networks. The consistency of this finding across both real and simulated settings strengthens the recommendation to avoid LL as a scoring function in hybrid BN structure learning. In terms of computational cost, both PCTABU and PCHC require similar running times across all simulated

configurations, confirming that the adoption of TABU in place of HC does not introduce a meaningful computational overhead for the network sizes examined here.

4. Conclusions

BNs are widely used across a variety of domains for modeling probabilistic and causal relationships. In fields like economics, it is especially important to build well-performing BNs to catch and evaluate the causal relations among variables. Therefore, just like in other kinds of models, the estimation performance of a BN is very important. BN models are constructed by using BN learning algorithms. Among these algorithms, hybrid BNs mainly depend on metrics called hyper-parameters and scoring functions. To build a new BN learning algorithm, the choice of these metrics is important in terms of the performance of the proposed algorithm. In this article, in addition to introducing a novel hybrid BN algorithm named PCTABU, the estimation performances of PCHC and PCTABU algorithms, which were built based on two different hyper-parameters HC and TABU, were compared. Moreover, the comparison was made using three different scoring functions named LL, BDe, and BIC. Consequently, the study demonstrated that both the choices of hyper-parameters and scoring functions affect the performances of BN algorithms.

Although two hyper-parameters and three scoring functions were employed in this article, the number of existing hyper-parameters and scoring functions is large. Therefore, it is possible to develop new algorithms by exploring different combinations of these components and comparing the resulting algorithms. In addition, a greater number of real-world-data-based BNs and higher-dimensional simulated data sets may also be employed as a continuation of this study.

Given that PCHC was previously shown to outperform MMHC on categorical data (Tsagris, 2021), and given that PCTABU achieves accuracy at least as high as PCHC in the present study, one would expect PCTABU to compare favorably against MMHC as well. A direct empirical comparison is left for future work, though the existing evidence provides a reasonable basis for this expectation.

Declarations and Statements

Availability of data and materials: The real-world-data based Bayesian networks used in this study are publicly available in the bnlearn Bayesian Network Repository at the following link: <https://www.bnlearn.com/bnrepository/> Both the PCHC and PCTABU algorithms are available in the *R* package *pchc* at the following links: <https://CRAN.R-project.org/package=pchc> and <https://doi.org/10.32614/CRAN.package.pchc>

Ethical approval: This article does not contain any studies with human participants or animals.

Funding: No funding was received from any individual, institution, or organization for this study.

Conflict of interest: The authors declare that there is no conflict of interest related to this study.

Author Contributions: The authors contributed equally to the preparation of this manuscript.

References

484 Acid, S., & De Campos, L. M. (2001). A hybrid methodology for learning belief networks:
 485 BENEDICT. *International Journal of Approximate Reasoning*, 27(3), 235–262.
 486 [https://doi.org/10.1016/S0888-613X\(01\)00041-X](https://doi.org/10.1016/S0888-613X(01)00041-X)

487 Chickering, D. M. (1996). Learning Bayesian networks is NP-complete. In *Learning from data:*
 488 *Artificial intelligence and statistics V* (pp. 121-130). New York, NY: Springer New York.
 489 https://doi.org/10.1007/978-1-4612-2404-4_12

490 Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of*
 491 *Machine Learning Research*, 3(Nov), 507-554.

492 Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic
 493 networks from data. *Machine Learning*, 9(4), 309–347.
 494 <https://doi.org/10.1007/BF00994110>

495 Cugnata, F., Kenett, R., & Salini, S. (2014). Bayesian network applications to customer surveys
 496 and InfoQ. *Procedia Economics and Finance*, 17, 3–9. [https://doi.org/10.1016/S2212-](https://doi.org/10.1016/S2212-5671(14)00871-5)
 497 [5671\(14\)00871-5](https://doi.org/10.1016/S2212-5671(14)00871-5)

498 Dash, D., & Druzdzel, M. J. (1999, July 30–August 1). A hybrid anytime algorithm for the
 499 construction of causal models from sparse data. Paper presented at the Fifteenth
 500 Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden.

501 De Campos, L. M., & Friedman, N. (2006). A scoring function for learning Bayesian networks
 502 based on mutual information and conditional independence tests. *Journal of Machine*
 503 *Learning Research*, 7(10).

504 Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine*
505 *Learning*, 29, 131–163. <https://doi.org/10.1023/A:1007465528199>

506 Friedman, N., Nachman, I., & Pe'er, D. (1999). Learning Bayesian network structure from
507 massive datasets: The "sparse candidate" algorithm. In *Fifteenth Conference on*
508 *Uncertainty in Artificial Intelligence (UAI-99)*.

509 Gemela, J. (2001). Financial analysis using Bayesian networks. *Applied Stochastic Models in*
510 *Business and Industry*, 17(1), 57–67. <https://doi.org/10.1002/asmb.422>

511 Glover, F. (1986). Future paths for integer programming and links to artificial intelligence.
512 *Computers & Operations Research*, 13(5), 533–549. <https://doi.org/10.1016/0305->
513 [0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1)

514 Glymour, C. (2001). *The Mind's Arrows: Bayes Nets and Graphical Causal Models in*
515 *Psychology*. Cambridge: MIT Press.

516 Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The
517 combination of knowledge and statistical data. *Machine Learning*, 20, 197–243.
518 <https://doi.org/10.1007/BF00994016>

519 Heckman, J. J. (2008). Econometric causality. *International Statistical Review*, 76(1), 1–27.
520 <https://doi.org/10.1111/j.1751-5823.2007.00024.x>

521 Hoover, K. D. (2008). Causality in economics and econometrics. In *The New Palgrave*
522 *Dictionary of Economics*. London: Palgrave Macmillan.

523 Kalisch, M., Maechler, M., Colombo, D., Maathuis, M. H., & Buehlmann, P. (2012). Causal
 524 inference using graphical models with the *R* package *pcalg*. *Journal of Statistical*
 525 *Software*, 47(11). <https://doi.org/10.18637/jss.v047.i11>

526 Margaritis, D. (2003). *Learning Bayesian network model structure from data* (Doctoral
 527 dissertation). School of Computer Science, Carnegie Mellon University.

528 Neapolitan, R. E. (2004). *Learning Bayesian Networks* (Vol. 38, p. 674). Upper Saddle River,
 529 NJ: Pearson Prentice Hall.

530 Papadakis, M., Tsagris, M., Fafalios, S., & Dimitriadis, M. (2026). *Rfast2: A collection of*
 531 *efficient and extremely fast R functions II* (R package version 0.1.5.6). [https://CRAN.R-](https://CRAN.R-project.org/package=Rfast2)
 532 [project.org/package=Rfast2](https://CRAN.R-project.org/package=Rfast2)

533 Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible*
 534 *Reasoning*. Los Altos: Morgan Kaufmann Publishers.

535 Rissanen, J. (1986). Stochastic complexity and modeling. *Annals of Statistics*, 14(3), 1080–1100.
 536 <https://doi.org/10.1214/aos/1176350051>

537 Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6, 461–464.
 538 <https://doi.org/10.1214/aos/1176344136>

539 Scutari, M., & Denis, J. B. (2021). *Bayesian Networks: With Examples in R*. Boca Raton: CRC
 540 Press. <https://doi.org/10.1201/9780429347436>

541 Singh, M., & Valtorta, M. (1995). Construction of Bayesian network structures from data: A
 542 brief survey and an efficient algorithm. *International Journal of Approximate Reasoning*,
 543 12(2), 111–131. [https://doi.org/10.1016/0888-613X\(94\)00016-V](https://doi.org/10.1016/0888-613X(94)00016-V)

544 Spirtes, P., & Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social*
 545 *Science Computer Review*, 9(1), 67–72. <https://doi.org/10.1177/089443939100900106>

546 Spirtes, P., Glymour, C. N., & Scheines, R. (2000). *Causation, Prediction, and Search*.
 547 Cambridge: MIT Press. <https://doi.org/10.7551/mitpress/1754.001.0001>

548 Teye, A. L., & Ahelegbey, D. F. (2017). Detecting spatial and temporal house price diffusion in
 549 the Netherlands: A Bayesian network approach. *Regional Science and Urban Economics*,
 550 65, 56–64. <https://doi.org/10.1016/j.regsciurbeco.2017.04.005>

551 Tsagris, M. (2019). Bayesian network learning with the PC algorithm: an improved and correct
 552 variation. *Applied Artificial Intelligence*, 33(2), 101–123.
 553 <https://doi.org/10.1080/08839514.2018.1526760>

554 Tsagris, M. (2021). A new scalable Bayesian network learning algorithm with applications to
 555 economics. *Computational Economics*, 57(1), 341–367. [https://doi.org/10.1007/s10614-](https://doi.org/10.1007/s10614-020-10065-7)
 556 [020-10065-7](https://doi.org/10.1007/s10614-020-10065-7)

557 Tsagris, M. (2026). *pchc: BN learning with the PCHC and related algorithms* (R package
 558 version 1.4). <https://CRAN.R-project.org/package=pchc>.
 559 <https://doi.org/10.32614/CRAN.package.pchc>

Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65, 31–78.
<https://doi.org/10.1007/s10994-006-6889-7>

Yaramakala, S., & Margaritis, D. (2005, November). Speculative Markov blanket discovery for optimal feature selection. Paper presented at the Fifth IEEE International Conference on Data Mining (ICDM'05). <https://doi.org/10.1109/ICDM.2005.134>

Appendix A

Comparative F-Score and SHD performance metrics tables of PCHC and PCTABU algorithms for the experimented BNs are provided below.

Table A1a. Comparative metrics of PCHC and PCTABU algorithms built for “child” data set.

	Number of Additions						Number of Deletions					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=1000	5.93	3.96	5.82	6.55	3.40	6.57	13.48	14.44	13.89	11.50	14.66	11.52
n=10000	2.88	3.06	3.05	2.36	2.51	2.37	7.36	8.82	8.11	3.80	8.97	3.81
n=20000	1.92	2.15	1.66	0.07	0.80	0.07	5.61	6.75	4.91	0.12	9.40	0.12
n=50000	2.02	2.07	1.97	0	0.33	0	5.87	5.85	5.68	0	5.84	0
n=1×10⁵	2.05	2.23	2.04	0	0.20	0	5.79	6.20	5.73	0	1.67	0
n=2×10⁵	1.98	2.15	1.98	0.03	0.07	0.03	5.51	5.95	5.55	0.10	0.34	0.10
n=5×10⁵	1.68	2.05	2.07	0	0.01	0	4.60	5.70	5.81	0	0.01	0
n=1×10⁶	1.92	1.98	2.07	0	0.03	0	5.35	5.47	5.72	0	0	0
n=2×10⁶	2.10	1.98	1.89	0	0.03	0	5.80	5.45	5.29	0	0.02	0
n=5×10⁶	1.89	1.82	1.89	0	0.02	0	5.26	5.07	5.32	0	0	0
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC

n=1000	7.86	7.67	7.77	6.55	7.18	6.57	27.27	26.07	27.48	24.60	25.24	24.66
n=10000	7.53	9.10	8.51	3.40	8.51	3.41	17.77	20.98	19.67	9.56	19.99	9.59
n=20000	7.59	8.80	6.60	0.09	10.09	0.09	15.12	17.70	13.17	0.28	20.29	0.28
n=50000	7.94	7.92	7.69	0	6.17	0	15.83	15.84	15.34	0	12.34	0
n=1×10⁵	7.86	8.41	7.77	0	1.85	0	15.70	16.84	15.54	0	3.72	0
n=2×10⁵	7.49	8.08	7.53	0.13	0.40	0.13	14.98	16.18	15.06	0.26	0.81	0.26
n=5×10⁵	6.28	7.74	7.88	0	0.01	0	12.56	15.49	15.76	0	0.03	0
n=1×10⁶	7.27	7.42	7.79	0	0	0	14.54	14.87	15.58	0	0.03	0
n=2×10⁶	7.90	7.40	7.18	0	0.02	0	15.80	14.83	14.36	0	0.07	0
n=5×10⁶	7.15	6.87	7.21	0	0	0	14.30	13.76	14.42	0	0.02	0

571 **Table A1b.** Comparative metrics of PCHC and PCTABU algorithms built for “child” data set.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=1000	0.42	0.40	0.41	0.51	0.41	0.51	0.53	0.56	0.52	0.59	0.59	0.59
n=10000	0.60	0.52	0.55	0.81	0.53	0.81	0.66	0.60	0.62	0.83	0.63	0.83
n=20000	0.64	0.58	0.69	0.99	0.47	0.99	0.70	0.65	0.74	0.99	0.61	0.99
n=50000	0.63	0.63	0.64	1	0.68	1	0.68	0.68	0.69	1	0.76	1
n=1×10⁵	0.63	0.60	0.64	1	0.91	1	0.69	0.66	0.69	1	0.93	1
n=2×10⁵	0.65	0.62	0.65	0.99	0.98	0.99	0.70	0.68	0.70	0.99	0.98	0.99
n=5×10⁵	0.71	0.64	0.63	1	1.00	1	0.75	0.69	0.69	1	1.00	1
n=1×10⁶	0.66	0.65	0.64	1	1	1	0.71	0.70	0.69	1	1.00	1
n=2×10⁶	0.63	0.65	0.66	1	1.00	1	0.68	0.70	0.71	1	1.00	1
n=5×10⁶	0.67	0.68	0.66	1	1	1	0.71	0.72	0.71	1	1.00	1
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=1000	0.47	0.47	0.46	0.55	0.48	0.55	0.21	0.21	0.21	0.21	0.21	0.21
n=10000	0.63	0.55	0.58	0.82	0.57	0.82	0.29	0.28	0.28	0.28	0.28	0.28
n=20000	0.67	0.61	0.71	0.99	0.53	0.99	0.24	0.24	0.23	0.23	0.23	0.23
n=50000	0.65	0.65	0.66	1	0.71	1	0.55	0.54	0.54	0.55	0.55	0.55
n=1×10⁵	0.66	0.63	0.66	1	0.91	1	1.01	1.00	1.00	1.01	1.01	1.00
n=2×10⁵	0.67	0.65	0.67	0.99	0.98	0.99	1.99	1.98	1.98	2.00	2	2.00
n=5×10⁵	0.73	0.66	0.66	1	1.00	1	4.99	4.97	4.96	5.02	5.01	4.99
n=1×10⁶	0.68	0.68	0.66	1	1	1	10.19	10.13	10.11	10.24	10.21	10.20
n=2×10⁶	0.66	0.68	0.69	1	1.00	1	20.45	20.43	20.37	20.56	20.52	20.55
n=5×10⁶	0.69	0.70	0.68	1	1	1	54.05	52.66	52.64	53.70	53.03	53.06

572 **Table A2a.** Comparative metrics of PCHC and PCTABU algorithms built for “insurance” data
573 set.

	Number of Additions						Number of Deletions					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	2.17	1.89	2.15	2.17	1.89	2.17	28.18	28.46	28.05	28.18	28.46	28.13
n=20000	2.35	2.33	2.39	2.39	2.39	2.38	27.02	27.24	26.84	27.74	28.29	26.99
n=50000	2.86	3.12	1.98	2.94	3.16	2	26.14	26.33	26.49	25.90	25.77	26.19
n=1×10⁵	1.98	1.98	1.96	2	2	2	24.93	24.99	24.81	24.25	24.25	24.25
n=2×10⁵	2	1.94	2	2	2	2	21.59	21.41	21.57	21.96	21.96	21.96
n=5×10⁵	1.98	1.90	1.98	2	2	2	18.88	18.40	18.88	19	19	19
n=1×10⁶	1.98	1.91	2	2	1.99	2	17.35	17.10	18.47	17.47	17.51	18.47
n=2×10⁶	1.98	1.94	2	2	2	2	16.83	16.94	17	17	17	17
n=5×10⁶	2.10	2.10	2.12	2.12	2.12	2.12	16.57	16.57	16.74	16.74	16.74	16.74
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	6.94	7.22	6.82	6.94	7.22	6.92	37.29	37.57	37.02	37.29	37.57	37.22
n=20000	7.98	8.20	7.84	8.74	9.29	7.98	37.35	37.77	37.07	38.87	39.97	37.35
n=50000	11.47	11.70	11.88	11.31	11.18	11.60	40.47	41.15	40.35	40.15	40.11	39.79
n=1×10⁵	11.66	11.72	11.52	11	11	11	38.57	38.69	38.29	37.25	37.25	37.25
n=2×10⁵	13.87	13.63	13.85	14.24	14.24	14.24	37.46	36.98	37.42	38.20	38.20	38.20
n=5×10⁵	13.86	13.30	13.86	14	14	14	34.72	33.60	34.72	35	35	35
n=1×10⁶	16.81	16.50	16.95	16.95	16.99	16.95	36.14	35.51	37.42	36.42	36.49	37.42
n=2×10⁶	18.26	18.33	18.45	18.45	18.45	18.45	37.07	37.21	37.45	37.45	37.45	37.45
n=5×10⁶	18.55	18.55	18.74	18.74	18.74	18.74	37.22	37.22	37.60	37.60	37.60	37.60

574 **Table A2b.** Comparative metrics of PCHC and PCTABU algorithms built for “insurance” data
575 set.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.47	0.46	0.47	0.47	0.46	0.47	0.77	0.77	0.78	0.77	0.77	0.77
n=20000	0.47	0.46	0.47	0.45	0.43	0.47	0.74	0.74	0.75	0.72	0.71	0.74
n=50000	0.43	0.42	0.42	0.44	0.44	0.43	0.66	0.65	0.67	0.67	0.67	0.67
n=1×10⁵	0.45	0.44	0.45	0.47	0.47	0.47	0.68	0.68	0.69	0.70	0.70	0.70

$n=2 \times 10^5$	0.46	0.47	0.46	0.45	0.45	0.45	0.66	0.66	0.66	0.65	0.65	0.65
$n=5 \times 10^5$	0.50	0.52	0.50	0.50	0.50	0.50	0.68	0.69	0.68	0.67	0.67	0.67
$n=1 \times 10^6$	0.48	0.49	0.46	0.48	0.48	0.46	0.63	0.64	0.62	0.62	0.62	0.62
$n=2 \times 10^6$	0.47	0.47	0.46	0.46	0.46	0.46	0.60	0.60	0.60	0.60	0.60	0.60
$n=5 \times 10^6$	0.47	0.47	0.46	0.46	0.46	0.46	0.60	0.60	0.59	0.59	0.59	0.59
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	0.58	0.57	0.59	0.58	0.57	0.58	0.12	0.11	0.10	0.10	0.10	0.10
$n=20000$	0.57	0.57	0.58	0.55	0.54	0.57	0.21	0.21	0.21	0.20	0.20	0.20
$n=50000$	0.52	0.51	0.51	0.53	0.53	0.52	0.49	0.49	0.49	0.48	0.48	0.48
$n=1 \times 10^5$	0.54	0.54	0.54	0.56	0.56	0.56	0.97	0.97	0.97	0.95	0.95	0.95
$n=2 \times 10^5$	0.54	0.55	0.54	0.53	0.53	0.53	1.94	1.93	1.92	1.92	1.91	1.91
$n=5 \times 10^5$	0.58	0.59	0.58	0.57	0.57	0.57	4.72	4.69	4.70	4.62	4.62	4.61
$n=1 \times 10^6$	0.55	0.55	0.53	0.54	0.54	0.53	9.72	9.68	9.69	9.64	9.61	9.64
$n=2 \times 10^6$	0.53	0.53	0.52	0.52	0.52	0.52	20.07	19.78	19.76	20.06	19.66	19.91
$n=5 \times 10^6$	0.52	0.52	0.52	0.52	0.52	0.52	56.79	52.77	52.73	55.28	52.48	53.23

576 **Table A3a.** Comparative metrics of PCHC and PCTABU algorithms built for “pigs” data set.

	Number of Additions						Number of Deletions					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	37.10	37.60	37.50	0	0.40	0	0	0	0	0	0	0
$n=20000$	40	41.20	40.80	0	0.40	0	0	0	0	0	0	0
$n=50000$	40.80	42.40	41.20	0	0.40	0	0	0	0	0	0	0
$n=1 \times 10^5$	43	43.80	42.40	0	0.60	0	0	0	0	0	0	0
$n=2 \times 10^5$	40.40	41.20	40.90	0	0.40	0	0	0	0	0	0	0
$n=5 \times 10^5$	41.80	41.80	41.80	0	0.40	0	0	0	0	0	0	0
$n=1 \times 10^6$	40.80	41.40	39.90	0	0.40	0	0	0	0	0	0	0
$n=2 \times 10^6$	40.60	41.30	40.80	0	0.80	0	0	0	0	0	0	0
$n=5 \times 10^6$	44.60	45.70	45.50	0	0.40	0	0	0	0	0	0	0
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	38	38.60	38.40	0	0	0	75.10	76.20	75.90	0	0.40	0
$n=20000$	40.60	41.80	41.40	0	0.20	0	80.60	83	82.20	0	0.60	0
$n=50000$	41	42.40	41.40	0	0	0	81.80	84.80	82.60	0	0.40	0
$n=1 \times 10^5$	43.40	44	42.80	0	0.20	0	86.40	87.80	85.20	0	0.80	0
$n=2 \times 10^5$	40.60	41.60	41.20	0	0.20	0	81	82.80	82.10	0	0.60	0
$n=5 \times 10^5$	43.80	43.60	43.80	0	0.20	0	85.60	85.40	85.60	0	0.60	0

$n=1 \times 10^6$	41.20	41.60	40.20	0	0	0	82	83	80.10	0	0.40	0
$n=2 \times 10^6$	41.40	41.60	41.60	0	0.20	0	82	82.90	82.40	0	1	0
$n=5 \times 10^6$	47.20	48	48	0	0.20	0	91.80	93.70	93.50	0	0.60	0

577 **Table A3b.** Comparative metrics of PCHC and PCTABU algorithms built for “pigs” data set.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	0.94	0.94	0.94	1	1	1	0.88	0.88	0.88	1	1.00	1
$n=20000$	0.93	0.93	0.93	1	1	1	0.87	0.87	0.87	1	1.00	1
$n=50000$	0.93	0.93	0.93	1	1	1	0.87	0.87	0.87	1	1.00	1
$n=1 \times 10^5$	0.93	0.93	0.93	1	1	1	0.86	0.86	0.87	1	1.00	1
$n=2 \times 10^5$	0.93	0.93	0.93	1	1	1	0.87	0.87	0.87	1	1.00	1
$n=5 \times 10^5$	0.93	0.93	0.93	1	1	1	0.86	0.86	0.86	1	1.00	1
$n=1 \times 10^6$	0.93	0.93	0.93	1	1	1	0.87	0.87	0.87	1	1.00	1
$n=2 \times 10^6$	0.93	0.93	0.93	1	1	1	0.87	0.87	0.87	1	1.00	1
$n=5 \times 10^6$	0.92	0.92	0.92	1	1	1	0.86	0.85	0.85	1	1.00	1
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	0.91	0.91	0.91	1	1	1	49.22	49.70	50.01	23.06	23.04	23.03
$n=20000$	0.90	0.90	0.90	1	1.00	1	68.40	69.26	69.35	41.10	41.07	41.15
$n=50000$	0.90	0.90	0.90	1	1	1	127.03	128.53	127.69	99.49	99.36	99.48
$n=1 \times 10^5$	0.89	0.89	0.90	1	1.00	1	220.12	220.86	220.90	192.32	192.45	192.48
$n=2 \times 10^5$	0.90	0.90	0.90	1	1.00	1	365.62	364.91	364.97	340.31	340.37	340.15
$n=5 \times 10^5$	0.89	0.90	0.89	1	1.00	1	874.44	874.81	873.75	848.43	848.69	848.66
$n=1 \times 10^6$	0.90	0.90	0.90	1	1	1	1744.23	1744.56	1742.78	1718.90	1718.50	1718.24
$n=2 \times 10^6$	0.90	0.90	0.90	1	1.00	1	3519.97	3520.58	3519.37	3496.30	3495.54	3495.03
$n=5 \times 10^6$	0.89	0.89	0.89	1	1.00	1	8880.58	8879.86	8878.38	8859.72	8858.10	8857.44

578 **Table A4a.** Comparative metrics of PCHC and PCTABU algorithms built for “alarm” data set.

	Number of Additions						Number of Deletions					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	11.41	11.42	11.86	7.04	7.07	7.10	8.92	8.96	9.07	8.55	8.50	8.60
$n=20000$	12.64	11.30	12.56	6.82	6.82	7.07	6.69	6.61	6.73	5.43	5.43	5.46
$n=50000$	10.91	9.82	11.60	6.02	6.21	6.01	5.61	5.38	5.87	4.04	4.05	4.04
$n=1 \times 10^5$	9.87	9.82	9.89	6.09	6.21	6.09	5.19	5.36	5.22	3.98	3.98	3.98

$n=2 \times 10^5$	9.93	9.92	9.88	6.13	6.23	6.13	4.81	4.94	4.99	3.61	3.62	3.61
$n=5 \times 10^5$	10.01	10.24	10.27	9	9.03	9	2.52	2.51	2.53	1.15	1.15	1.15
$n=1 \times 10^6$	10.21	10.35	10.28	9.06	9.11	9.02	2.47	2.42	2.35	1.02	1.04	1.02
$n=2 \times 10^6$	10.41	10.26	10.52	9.16	9.25	9.12	2.16	2.23	2.27	0.92	0.93	0.92
$n=5 \times 10^6$	10.88	11.37	11.33	9.72	10.33	10.23	1.39	1.27	1.43	0.02	0.08	0.02
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	14.22	14.46	15.47	7.92	7.86	8.28	34.55	34.84	36.40	23.51	23.43	23.98
$n=20000$	17.11	16.56	17.07	6.86	6.79	7.16	36.44	34.47	36.36	19.11	19.04	19.69
$n=50000$	16.45	16.27	17.13	6	6.12	6	32.97	31.47	34.60	16.06	16.38	16.05
$n=1 \times 10^5$	15.92	16.61	16.08	6.10	6.18	6.12	30.98	31.79	31.19	16.17	16.37	16.19
$n=2 \times 10^5$	16.46	17.09	17.43	6.46	6.57	6.46	31.20	31.95	32.30	16.20	16.42	16.20
$n=5 \times 10^5$	18.30	18.47	18.63	9	9	9	30.83	31.22	31.43	19.15	19.18	19.15
$n=1 \times 10^6$	19.10	18.78	18.45	9.08	9.05	9.08	31.78	31.55	31.08	19.16	19.20	19.12
$n=2 \times 10^6$	18.13	18.22	18.77	9.22	9.22	9.18	30.70	30.71	31.56	19.30	19.40	19.22
$n=5 \times 10^6$	19.11	18.70	19.76	9.60	10.11	10.12	31.38	31.34	32.52	19.34	20.52	20.37

579 **Table A4b.** Comparative metrics of PCHC and PCTABU algorithms built for “alarm” data set.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	0.54	0.53	0.51	0.67	0.67	0.66	0.51	0.51	0.48	0.69	0.69	0.68
$n=20000$	0.52	0.54	0.52	0.75	0.76	0.75	0.47	0.49	0.47	0.73	0.74	0.72
$n=50000$	0.56	0.57	0.54	0.80	0.80	0.80	0.51	0.52	0.48	0.77	0.76	0.77
$n=1 \times 10^5$	0.58	0.56	0.57	0.80	0.80	0.80	0.53	0.51	0.52	0.77	0.76	0.77
$n=2 \times 10^5$	0.57	0.56	0.55	0.80	0.80	0.80	0.52	0.51	0.50	0.76	0.76	0.76
$n=5 \times 10^5$	0.58	0.58	0.58	0.80	0.80	0.80	0.51	0.50	0.50	0.69	0.69	0.69
$n=1 \times 10^6$	0.57	0.58	0.58	0.80	0.80	0.80	0.49	0.50	0.50	0.69	0.69	0.69
$n=2 \times 10^6$	0.59	0.59	0.58	0.80	0.80	0.80	0.51	0.51	0.50	0.69	0.68	0.69
$n=5 \times 10^6$	0.59	0.60	0.58	0.81	0.80	0.80	0.50	0.50	0.48	0.68	0.66	0.66
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	0.53	0.52	0.50	0.68	0.68	0.67	0.19	0.18	0.18	0.17	0.17	0.17
$n=20000$	0.49	0.51	0.49	0.74	0.74	0.74	0.34	0.34	0.34	0.33	0.33	0.33
$n=50000$	0.53	0.54	0.51	0.78	0.78	0.78	0.82	0.81	0.81	0.80	0.80	0.80
$n=1 \times 10^5$	0.55	0.54	0.55	0.78	0.78	0.78	1.66	1.65	1.64	1.64	1.64	1.63
$n=2 \times 10^5$	0.55	0.53	0.53	0.78	0.78	0.78	3.34	3.31	3.31	3.34	3.32	3.33
$n=5 \times 10^5$	0.54	0.54	0.54	0.74	0.74	0.74	8.78	8.75	8.76	8.85	8.83	8.84

$n=1 \times 10^6$	0.53	0.53	0.54	0.74	0.74	0.74	18.02	17.99	17.97	18.19	18.17	18.14
$n=2 \times 10^6$	0.55	0.55	0.54	0.74	0.74	0.74	40.50	40.41	40.41	40.88	40.83	40.81
$n=5 \times 10^6$	0.54	0.55	0.53	0.74	0.72	0.72	102.18	99.71	99.60	101.11	100.71	100.67

580 **Table A5a.** Comparative metrics of PCHC and PCTABU algorithms built for “hailfinder” data
581 set.

	Number of Additions						Number of Deletions					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	11.08	12.63	10.65	9.36	10.05	8.84	48.58	48.33	48.54	48.92	48.33	49.04
$n=20000$	11.15	12.66	10.15	9.74	10.18	9.55	47.95	47.94	47.94	47.94	47.94	47.94
$n=50000$	13.32	13	13.21	8.46	10.25	8.81	47.65	47.65	47.65	47.65	47.65	47.65
$n=1 \times 10^5$	13.21	12.86	13.43	9.58	9.42	10.02	47.20	47.18	47.18	47.19	47.18	47.18
$n=2 \times 10^5$	13.64	13.53	14.06	7.26	8.91	9.01	47	47	47	47.01	47	47.01
$n=5 \times 10^5$	13.05	13.72	13.14	6.71	8.65	6.70	47	47	47	47	47	47
$n=1 \times 10^6$	12.66	14	12.98	7.19	9.22	6.84	47	47	47	47	47	47
$n=2 \times 10^6$	14.70	15.26	14.62	8.75	9.84	8.69	47	47	47	47	47	47
$n=5 \times 10^6$	13.91	15.42	14.76	9.12	11.42	8.68	47	47	47	47	47	47
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	12.46	13.15	12.17	10.01	10.66	9.45	72.12	74.11	71.36	68.29	69.04	67.33
$n=20000$	12.05	12.40	12.29	10.75	9.99	11.93	71.15	73	70.38	68.43	68.11	69.42
$n=50000$	13.01	12.47	13.02	8.67	9.96	8.93	73.98	73.12	73.88	64.78	67.86	65.39
$n=1 \times 10^5$	12.87	12.47	12.83	9.37	8.48	9.60	73.28	72.51	73.44	66.14	65.08	66.80
$n=2 \times 10^5$	12.24	12	12.37	5.80	6.04	7.12	72.88	72.53	73.43	60.07	61.95	63.14
$n=5 \times 10^5$	11.69	11.89	11.89	4.66	5.18	4.44	71.74	72.61	72.03	58.37	60.83	58.14
$n=1 \times 10^6$	11.26	12.14	11.69	4.78	5.50	4.97	70.92	73.14	71.67	58.97	61.72	58.81
$n=2 \times 10^6$	12.108	11.308	11.902	5.73	5.33	5.63	73.808	73.604	73.504	61.48	62.17	61.32
$n=5 \times 10^6$	11.08	11.60	11.75	6.39	5.50	5.41	71.99	74.02	73.51	62.51	63.92	61.09

582 **Table A5b.** Comparative metrics of PCHC and PCTABU algorithms built for “hailfinder” data
583 set.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	0.27	0.26	0.27	0.29	0.29	0.29	0.49	0.46	0.50	0.56	0.54	0.57

n=20000	0.28	0.27	0.27	0.29	0.30	0.28	0.50	0.48	0.51	0.54	0.55	0.52
n=50000	0.27	0.28	0.27	0.32	0.31	0.32	0.46	0.47	0.46	0.61	0.56	0.60
n=1×10⁵	0.28	0.28	0.28	0.32	0.33	0.32	0.47	0.48	0.47	0.59	0.61	0.57
n=2×10⁵	0.29	0.29	0.28	0.36	0.36	0.35	0.48	0.49	0.48	0.71	0.67	0.65
n=5×10⁵	0.29	0.29	0.29	0.38	0.37	0.38	0.50	0.49	0.49	0.74	0.69	0.74
n=1×10⁶	0.30	0.29	0.29	0.38	0.37	0.37	0.51	0.48	0.50	0.73	0.68	0.73
n=2×10⁶	0.29	0.30	0.29	0.36	0.37	0.37	0.47	0.48	0.48	0.68	0.67	0.68
n=5×10⁶	0.30	0.29	0.29	0.36	0.37	0.37	0.50	0.48	0.48	0.66	0.64	0.69
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.34	0.33	0.35	0.38	0.38	0.39	0.22	0.22	0.22	0.19	0.20	0.19
n=20000	0.36	0.35	0.36	0.38	0.39	0.36	0.42	0.42	0.42	0.38	0.38	0.38
n=50000	0.34	0.35	0.34	0.42	0.40	0.42	0.97	0.97	0.96	0.93	0.92	0.93
n=1×10⁵	0.35	0.35	0.35	0.41	0.43	0.41	1.89	1.87	1.88	1.83	1.83	1.83
n=2×10⁵	0.36	0.36	0.36	0.48	0.47	0.45	3.95	3.91	3.93	3.91	3.92	3.89
n=5×10⁵	0.37	0.36	0.36	0.50	0.48	0.50	9.76	9.73	9.73	9.76	9.74	9.71
n=1×10⁶	0.38	0.36	0.37	0.50	0.48	0.49	19.42	19.46	19.42	19.38	19.49	19.45
n=2×10⁶	0.36	0.37	0.36	0.47	0.48	0.48	40.41	39.81	39.66	39.75	39.72	39.73
n=5×10⁶	0.38	0.36	0.36	0.46	0.47	0.48	102.30	101.03	101.00	101.07	101.51	101.16

584 **Table A6a.** Comparative metrics of PCHC and PCTABU algorithms built for “hepar2” data set.

	Number of Additions						Number of Deletions					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	15.90	15.85	15.90	15.95	15.25	15.35	73.40	73.15	73.70	73.50	74.30	73.75
n=20000	12.45	13.90	12.05	11.95	12.95	11.65	66.75	66.75	67.30	66.90	69.10	67.35
n=50000	5.50	8.95	5.50	5.45	7.80	5.50	60.90	61.25	61.05	60.95	62.65	61.10
n=1×10⁵	6.15	7.95	6.15	6.15	7.20	6.15	55.45	56.60	55.95	55.45	57.75	55.95
n=2×10⁵	6	8.05	6	6.10	7.50	6.10	50.90	51.40	50.90	50.90	53.60	50.90
n=5×10⁵	7.25	8.30	7.15	7.25	7.60	7.15	42.45	43.75	42.45	42.45	46.95	42.45
n=1×10⁶	7.40	7.55	6.40	7.45	6.60	6.50	38.50	38.25	38.25	38.25	43	38.25
n=2×10⁶	5.40	6.80	5.40	5.40	6.70	5.40	34.40	34.95	34.40	34.40	37.70	34.40
n=5×10⁶	5.30	5.60	5.30	5.30	5.20	5.30	23.55	24.05	23.55	23.55	26.05	23.55
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	20.85	20.10	20.60	21.10	21.60	20.35	110.15	109.10	110.20	110.55	111.15	109.45
n=20000	17.60	17.80	16.05	18.15	20.30	17	96.80	98.45	95.40	97	102.35	96
n=50000	10.75	12.10	10.75	10.80	13.50	10.75	77.15	82.30	77.30	77.20	83.95	77.35

$n=1 \times 10^5$	11.15	13.10	11	11.15	14.15	11	72.75	77.65	73.10	72.75	79.10	73.10
$n=2 \times 10^5$	11.05	11.60	11.05	11.15	13.50	11.15	67.95	71.05	67.95	68.15	74.60	68.15
$n=5 \times 10^5$	12.05	13.80	12.75	12.05	16.55	12.75	61.75	65.85	62.35	61.75	71.10	62.35
$n=1 \times 10^6$	12.10	12.30	11	11.90	16.45	11.10	58	58.10	55.65	57.60	66.05	55.85
$n=2 \times 10^6$	10.25	11.95	10.10	10.25	14.70	10.10	50.05	53.70	49.90	50.05	59.10	49.90
$n=5 \times 10^6$	9.25	10.10	9.25	9.25	11.90	9.25	38.10	39.75	38.10	38.10	43.15	38.10

585 **Table A6b.** Comparative metrics of PCHC and PCTABU algorithms built for “hepar2” data set.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	0.29	0.29	0.29	0.28	0.27	0.29	0.51	0.52	0.51	0.50	0.50	0.52
$n=20000$	0.36	0.36	0.37	0.36	0.32	0.36	0.61	0.60	0.64	0.61	0.56	0.63
$n=50000$	0.46	0.44	0.46	0.46	0.42	0.46	0.79	0.74	0.79	0.79	0.72	0.79
$n=1 \times 10^5$	0.49	0.47	0.49	0.49	0.46	0.49	0.79	0.75	0.79	0.79	0.74	0.79
$n=2 \times 10^5$	0.53	0.52	0.53	0.53	0.49	0.53	0.80	0.78	0.80	0.80	0.76	0.80
$n=5 \times 10^5$	0.59	0.56	0.58	0.59	0.52	0.58	0.80	0.77	0.79	0.80	0.74	0.79
$n=1 \times 10^6$	0.62	0.62	0.63	0.62	0.55	0.63	0.81	0.80	0.83	0.81	0.76	0.82
$n=2 \times 10^6$	0.66	0.65	0.66	0.66	0.60	0.66	0.85	0.82	0.85	0.85	0.79	0.85
$n=5 \times 10^6$	0.75	0.74	0.75	0.75	0.71	0.75	0.87	0.86	0.87	0.87	0.84	0.87
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
$n=10000$	0.37	0.38	0.37	0.36	0.35	0.37	0.32	0.34	0.33	0.26	0.26	0.26
$n=20000$	0.46	0.45	0.47	0.45	0.41	0.46	0.59	0.58	0.57	0.50	0.50	0.50
$n=50000$	0.58	0.55	0.58	0.58	0.53	0.58	1.29	1.31	1.30	1.20	1.22	1.20
$n=1 \times 10^5$	0.61	0.58	0.61	0.61	0.56	0.61	2.51	2.53	2.50	2.41	2.44	2.39
$n=2 \times 10^5$	0.64	0.62	0.64	0.64	0.59	0.64	4.90	4.94	4.92	4.81	4.82	4.81
$n=5 \times 10^5$	0.68	0.65	0.67	0.68	0.61	0.67	12.35	12.49	12.57	12.32	12.21	12.25
$n=1 \times 10^6$	0.70	0.70	0.71	0.70	0.64	0.71	26.07	26.19	26.18	25.92	26.24	25.85
$n=2 \times 10^6$	0.74	0.72	0.74	0.74	0.68	0.74	49.30	49.17	49.22	48.86	49.11	48.75
$n=5 \times 10^6$	0.80	0.79	0.80	0.80	0.77	0.80	135.76	134.43	134.57	133.04	133.68	132.41

586 **Table A7a.** Comparative metrics of PCHC and PCTABU algorithms built for “win95pts” data
587 set.

Number of Additions		Number of Deletions	
PCHC	PCTABU	PCHC	PCTABU

	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	9.40	9.70	9	9.50	9.60	8.85	45.70	45.95	45.60	45.80	46	45.60
n=20000	8.20	8.35	8	8.20	8.35	7.65	37.45	37.45	37.25	39.25	37.45	37.25
n=50000	7.45	7.40	7.25	7.55	7.40	7.25	32.50	32.30	32.15	34.15	32.30	32.15
n=1×10⁵	6.10	6.40	5.95	5.90	6.15	5.80	30.75	30.50	30.45	32.45	30.50	30.45
n=2×10⁵	5.55	5.70	5.35	5.45	5.75	5.20	29.30	28.90	28.90	30.90	28.90	28.90
n=5×10⁵	4.55	4.65	4.45	4	4.15	3.90	27.15	27.15	27.15	27.15	27.15	27.15
n=1×10⁶	5.30	5.50	5	4.50	4.75	4.10	24.10	24.15	24.10	24.10	24.15	24.10
n=2×10⁶	5.20	5.50	5.05	4.35	4.80	4.15	23.35	23.40	23.35	23.35	23.40	23.35
n=5×10⁶	5.30	5.60	5	4.50	4.95	4.10	22.10	22.10	22.10	22.10	22.10	22.10
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	15	15.70	15.05	15.55	15.75	15.40	70.10	71.35	69.65	70.85	71.35	69.85
n=20000	15.30	15.15	14.95	17.10	15.30	15.10	60.95	60.95	60.20	64.55	61.10	60
n=50000	16.05	15.80	15.65	17.65	15.80	15.65	56	55.50	55.05	59.35	55.50	55.05
n=1×10⁵	14.60	14.30	14.40	16.50	14.55	14.55	51.45	51.20	50.80	54.85	51.20	50.80
n=2×10⁵	14.90	14.55	14.50	16.60	14.60	14.65	49.75	49.15	48.75	52.95	49.25	48.75
n=5×10⁵	13.55	13.55	13.55	14.15	14.10	14.15	45.25	45.35	45.15	45.30	45.40	45.20
n=1×10⁶	10.45	10.50	10.45	11.25	11.25	11.35	39.85	40.15	39.55	39.85	40.15	39.55
n=2×10⁶	9.75	9.90	9.70	10.55	10.50	10.55	38.30	38.80	38.10	38.25	38.70	38.05
n=5×10⁶	9.65	9.70	9.65	10.45	10.30	10.55	37.05	37.40	36.75	37.05	37.35	36.75

588 **Table A7b.** Comparative metrics of PCHC and PCTABU algorithms built for “win95pts” data
589 set.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.51	0.50	0.51	0.51	0.50	0.51	0.72	0.71	0.72	0.71	0.71	0.72
n=20000	0.57	0.58	0.58	0.55	0.57	0.58	0.75	0.75	0.76	0.73	0.75	0.76
n=50000	0.61	0.61	0.61	0.58	0.61	0.61	0.76	0.77	0.77	0.74	0.77	0.77
n=1×10⁵	0.63	0.64	0.64	0.60	0.64	0.64	0.79	0.79	0.80	0.77	0.79	0.80
n=2×10⁵	0.64	0.65	0.65	0.62	0.65	0.65	0.80	0.80	0.80	0.78	0.80	0.80
n=5×10⁵	0.67	0.67	0.67	0.67	0.67	0.67	0.82	0.82	0.82	0.82	0.82	0.82
n=1×10⁶	0.72	0.72	0.72	0.71	0.71	0.71	0.85	0.85	0.85	0.85	0.85	0.85
n=2×10⁶	0.73	0.73	0.73	0.73	0.73	0.73	0.86	0.85	0.86	0.86	0.85	0.86
n=5×10⁶	0.74	0.74	0.74	0.74	0.74	0.74	0.86	0.86	0.86	0.86	0.86	0.86
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		

	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.60	0.59	0.60	0.59	0.59	0.60	0.35	0.35	0.35	0.26	0.24	0.25
n=20000	0.65	0.65	0.66	0.62	0.65	0.66	0.59	0.59	0.59	0.48	0.47	0.48
n=50000	0.68	0.68	0.68	0.65	0.68	0.68	1.30	1.28	1.29	1.16	1.14	1.14
n=1×10⁵	0.70	0.71	0.71	0.68	0.71	0.71	2.48	2.49	2.48	2.31	2.31	2.31
n=2×10⁵	0.71	0.72	0.72	0.69	0.72	0.72	4.74	4.75	4.74	4.54	4.54	4.53
n=5×10⁵	0.74	0.74	0.74	0.74	0.74	0.74	11.64	11.64	11.62	11.38	11.32	11.32
n=1×10⁶	0.78	0.78	0.78	0.78	0.78	0.78	23.48	23.41	23.42	23.01	23.00	23.02
n=2×10⁶	0.79	0.79	0.79	0.79	0.79	0.79	53.50	53.01	53.26	52.36	52.03	52.07
n=5×10⁶	0.80	0.80	0.80	0.79	0.79	0.79	131.16	120.71	120.67	119.31	118.93	119.13

590 **Table A8a.** Comparative metrics of PCHC and PCTABU algorithms built for “andes” data set.

	Number of Additions						Number of Deletions					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	11.75	11.65	12.25	11.75	11.65	12.25	70.75	70.60	70.75	70.80	70.60	70.80
n=20000	8.45	9.80	8.70	8.60	9.80	8.75	56.05	55.90	56	56.10	55.90	56.05
n=50000	7.65	9.55	6.75	7.65	9.55	6.80	36.35	36.10	36.15	36.35	36.10	36.35
n=1×10⁵	3.90	7.10	3.55	3.90	7.10	3.55	23.70	23.60	23.65	23.70	23.60	23.65
n=2×10⁵	2.60	6.60	2.60	2.60	6.60	2.60	13.15	13.15	13.15	13.15	13.15	13.15
n=5×10⁵	2.30	4.45	1.90	2.30	4.45	1.90	6.10	6.10	6.10	6.10	6.10	6.10
n=1×10⁶	1.05	3.85	0.95	1.05	3.85	0.95	5.60	5.60	5.60	5.60	5.60	5.60
n=2×10⁶	1.20	4.15	0.90	1.20	3.95	0.90	5.15	5.15	5.15	5.15	5.15	5.15
n=5×10⁶	0.50	4.40	0.35	0.50	4.40	0.35	5	5	5	5	5	5
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	3.75	4.45	3.85	3.75	4.45	3.85	86.25	86.70	86.85	86.30	86.70	86.90
n=20000	3.30	5.15	3.50	3.40	5.15	3.45	67.80	70.85	68.20	68.10	70.85	68.25
n=50000	1.85	2.65	1.65	1.85	2.75	1.85	45.85	48.30	44.55	45.85	48.40	45
n=1×10⁵	1.20	1.50	1.20	1.20	1.50	1.20	28.80	32.20	28.40	28.80	32.20	28.40
n=2×10⁵	0.70	0.70	0.70	0.70	0.70	0.70	16.45	20.45	16.45	16.45	20.45	16.45
n=5×10⁵	0.50	0.50	0.50	0.50	0.50	0.50	8.90	11.05	8.50	8.90	11.05	8.50
n=1×10⁶	0.55	0.65	0.55	0.55	0.65	0.55	7.20	10.10	7.10	7.20	10.10	7.10
n=2×10⁶	0.15	0.15	0.15	0.15	0.15	0.15	6.50	9.45	6.20	6.50	9.25	6.20
n=5×10⁶	0	0	0	0	0	0	5.50	9.40	5.35	5.50	9.40	5.35

591 **Table A8b.** Comparative metrics of PCHC and PCTABU algorithms built for “andes” data set.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=1000	0.53	0.52	0.53	0.53	0.52	0.53	0.82	0.82	0.82	0.82	0.82	0.82
n=10000	0.79	0.78	0.79	0.79	0.78	0.79	0.95	0.94	0.94	0.95	0.94	0.94
n=20000	0.83	0.82	0.83	0.83	0.82	0.83	0.96	0.95	0.96	0.96	0.95	0.96
n=50000	0.89	0.89	0.89	0.89	0.89	0.89	0.97	0.96	0.97	0.97	0.96	0.97
n=1×10⁵	0.93	0.93	0.93	0.93	0.93	0.93	0.98	0.97	0.99	0.98	0.97	0.99
n=2×10⁵	0.96	0.96	0.96	0.96	0.96	0.96	0.99	0.98	0.99	0.99	0.98	0.99
n=5×10⁵	0.98	0.98	0.98	0.98	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99
n=1×10⁶	0.98	0.98	0.98	0.98	0.98	0.98	0.99	0.99	1.00	0.99	0.99	1.00
n=2×10⁶	0.98	0.98	0.98	0.98	0.98	0.98	1.00	0.99	1.00	1.00	0.99	1.00
n=5×10⁶	0.99	0.99	0.99	0.99	0.99	0.99	1.00	0.99	1.00	1.00	0.99	1.00
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.86	0.86	0.86	0.86	0.86	0.86	3.81	3.86	3.79	0.89	0.88	0.88
n=20000	0.89	0.88	0.89	0.89	0.88	0.89	4.62	4.98	4.57	1.60	1.61	1.59
n=50000	0.93	0.92	0.93	0.93	0.92	0.93	7.01	7.30	6.94	3.81	3.81	3.79
n=1×10⁵	0.96	0.95	0.96	0.96	0.95	0.96	10.89	11.37	10.82	7.51	7.54	7.56
n=2×10⁵	0.97	0.97	0.97	0.97	0.97	0.97	18.32	18.72	18.21	14.79	14.78	14.75
n=5×10⁵	0.99	0.98	0.99	0.99	0.98	0.99	40.54	41.03	40.47	36.88	36.74	36.78
n=1×10⁶	0.99	0.98	0.99	0.99	0.98	0.99	79.30	77.90	77.15	73.55	73.33	73.33
n=2×10⁶	0.99	0.99	0.99	0.99	0.99	0.99	162.23	153.64	152.70	148.53	148.45	148.50
n=5×10⁶	0.99	0.99	0.99	0.99	0.99	0.99	413.83	385.02	384.17	379.37	378.56	378.24

592

593 Appendix B

594 Comparative F-Score and SHD performance metrics tables of PCHC and PCTABU algorithms
595 for the simulated BNs are provided below.

596 **Table B1a.** Comparative metrics of PCHC and PCTABU algorithms regarding the simulated BN
597 built for the data set with 70 nodes and 3 neighbors on average.

Number of Additions	Number of Deletions
---------------------	---------------------

	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	14.05	17.95	14.45	13.10	17.85	12.80	14	13.25	14	13.95	13.25	13.90
n=20000	14.25	18.90	13.50	12.80	18.75	12	12.35	11.85	12.35	12.35	11.85	12.20
n=50000	19.85	27.60	20.35	18.55	26.90	18.75	10.85	10.55	10.90	10.80	10.55	10.90
n=1×10⁵	20.95	27.85	20.10	19.75	27.20	19.55	9.30	9.20	9.40	9.35	9.20	9.35
n=2×10⁵	26.95	35.05	27.45	25.10	33.25	25.30	7.95	7.70	8.10	7.95	7.70	8.05
n=5×10⁵	28.90	35.50	28.75	27.95	35	27.85	10.05	9.75	10	10	9.75	10
n=1×10⁶	32.35	40.60	32.10	31.15	40.30	31.90	6.40	6.05	6.35	6.35	6.05	6.30
n=5×10⁶	39.95	47.30	39.70	39.30	47.80	38.85	4.45	4.30	4.45	4.45	4.30	4.40
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	11.25	10.20	10.95	10.80	10.45	10.65	39.30	41.40	39.40	37.85	41.55	37.35
n=20000	9.25	9.05	9.10	9.40	9.25	9.10	35.85	39.80	34.95	34.55	39.85	33.30
n=50000	9.20	10.60	9.25	8.85	10.65	8.90	39.90	48.75	40.50	38.20	48.10	38.55
n=1×10⁵	8.10	9.45	7.75	8	9.40	8.15	38.35	46.50	37.25	37.10	45.80	37.05
n=2×10⁵	8.90	10.50	9.15	8.80	10.30	8.75	43.80	53.25	44.70	41.85	51.25	42.10
n=5×10⁵	8.50	9.40	8.45	8.30	9.25	8.35	47.45	54.65	47.20	46.25	54	46.20
n=1×10⁶	8	9.20	7.55	7.50	9.15	7.55	46.75	55.85	46	45	55.50	45.75
n=5×10⁶	7.85	9.20	7.70	7.60	9.50	7.50	52.25	60.80	51.85	51.35	61.60	50.75

598 **Table B1b.** Comparative metrics of PCHC and PCTABU algorithms regarding the simulated BN
599 built for the data set with 70 nodes and 3 neighbors on average.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.46	0.50	0.47	0.47	0.49	0.47	0.47	0.46	0.47	0.48	0.46	0.49
n=20000	0.52	0.53	0.52	0.51	0.53	0.52	0.50	0.47	0.51	0.51	0.47	0.53
n=50000	0.56	0.53	0.55	0.57	0.53	0.56	0.47	0.39	0.46	0.48	0.39	0.48
n=1×10⁵	0.61	0.58	0.62	0.61	0.58	0.61	0.49	0.41	0.50	0.50	0.42	0.50
n=2×10⁵	0.61	0.57	0.60	0.61	0.58	0.61	0.43	0.35	0.42	0.44	0.37	0.44
n=5×10⁵	0.61	0.60	0.61	0.62	0.60	0.62	0.44	0.39	0.44	0.45	0.40	0.45
n=1×10⁶	0.69	0.67	0.70	0.70	0.67	0.70	0.45	0.39	0.46	0.46	0.39	0.46
n=5×10⁶	0.71	0.68	0.71	0.71	0.67	0.72	0.39	0.34	0.39	0.39	0.33	0.40
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.46	0.47	0.47	0.48	0.47	0.48	0.35	0.35	0.35	0.34	0.34	0.33

n=20000	0.51	0.50	0.51	0.51	0.49	0.52	0.83	0.82	0.82	0.81	0.81	0.80
n=50000	0.51	0.45	0.50	0.52	0.45	0.52	2.79	2.78	2.77	2.74	2.77	2.77
n=1×10⁵	0.54	0.48	0.55	0.55	0.49	0.55	7.31	7.40	7.35	7.36	7.34	7.34
n=2×10⁵	0.50	0.44	0.49	0.51	0.45	0.51	22.44	22.46	22.34	22.31	22.39	22.36
n=5×10⁵	0.51	0.47	0.51	0.52	0.47	0.52	170.91	171.11	171.09	171.08	170.85	170.92
n=1×10⁶	0.54	0.49	0.55	0.55	0.49	0.55	739.57	739.60	739.41	739.18	739.31	739.19
n=5×10⁶	0.50	0.45	0.50	0.51	0.44	0.51	6548.53	6533.25	6528.75	6522.29	6523.69	6521.68

600 **Table B2a.** Comparative metrics of PCHC and PCTABU algorithms regarding the simulated BN
601 built for the data set with 70 nodes and 5 neighbors on average.

	Number of Additions						Number of Deletions					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	18.85	23.75	19.50	17.15	22.70	17.40	40.45	39.65	40.50	40.45	39.65	40.40
n=20000	21.70	25.80	22.30	20.30	25.50	20.75	36.25	34.75	35.90	36.05	34.75	36
n=50000	24.35	28.50	24.40	22.90	27.90	23.50	34.85	33.70	34.80	34.80	33.70	34.65
n=1×10⁵	24.35	29.50	24.80	22.20	28.50	22.80	32.10	31.45	32.05	32.25	31.45	32.05
n=2×10⁵	27.80	34	27.95	26.10	32.90	26.55	31	30.40	30.95	31	30.40	31
n=5×10⁵	32.30	37.90	32.35	31.05	37.30	31.10	31.15	30.20	31.10	31.10	30.20	31.05
n=1×10⁶	36.90	43.35	37	35.40	42.10	35.85	26.95	26.35	26.90	26.95	26.35	26.90
n=5×10⁶	45.50	50.95	45.25	44.50	49.60	43.95	27.15	26.55	27.10	27.15	26.55	27.10
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	12.70	13.15	12.75	12.45	12.75	12.30	72	76.55	72.75	70.05	75.10	70.10
n=20000	12.10	11.65	12.25	12.25	11.85	12.05	70.05	72.20	70.45	68.60	72.10	68.80
n=50000	13.30	12.90	12.90	12.75	13.10	12.95	72.50	75.10	72.10	70.45	74.70	71.10
n=1×10⁵	10.45	9.90	10.30	9.85	9.80	10.25	66.90	70.85	67.15	64.30	69.75	65.10
n=2×10⁵	11.45	11.10	11.20	10.90	11.20	10.85	70.25	75.50	70.10	68	74.50	68.40
n=5×10⁵	10.30	10.50	10.10	10.20	10.35	10	73.75	78.60	73.55	72.35	77.85	72.15
n=1×10⁶	11	10.85	11.15	11.10	11.10	11.10	74.85	80.55	75.05	73.45	79.55	73.85
n=5×10⁶	12.15	12.05	11.85	11.80	11.75	11.60	84.80	89.55	84.20	83.45	87.90	82.65

602 **Table B2b.** Comparative metrics of PCHC and PCTABU algorithms regarding the simulated BN
603 built for the data set with 70 nodes and 5 neighbors on average.

Precision	Recall
-----------	--------

	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.30	0.30	0.30	0.30	0.31	0.30	0.42	0.38	0.41	0.44	0.40	0.44
n=20000	0.35	0.37	0.35	0.35	0.37	0.35	0.44	0.43	0.44	0.45	0.43	0.45
n=50000	0.36	0.39	0.37	0.37	0.38	0.37	0.42	0.41	0.43	0.44	0.41	0.44
n=1×10⁵	0.43	0.45	0.44	0.44	0.45	0.43	0.48	0.46	0.48	0.51	0.47	0.49
n=2×10⁵	0.45	0.46	0.46	0.46	0.46	0.46	0.47	0.44	0.47	0.49	0.45	0.49
n=5×10⁵	0.47	0.48	0.47	0.47	0.48	0.48	0.46	0.44	0.47	0.47	0.44	0.47
n=1×10⁶	0.50	0.51	0.49	0.49	0.50	0.50	0.44	0.41	0.44	0.45	0.42	0.45
n=5×10⁶	0.50	0.51	0.50	0.50	0.51	0.51	0.40	0.39	0.41	0.41	0.40	0.42
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.35	0.34	0.35	0.36	0.35	0.36	1.10	1.10	1.09	1.09	1.09	1.08
n=20000	0.39	0.40	0.39	0.39	0.40	0.39	1.10	1.11	1.10	1.08	1.08	1.09
n=50000	0.39	0.40	0.40	0.40	0.40	0.40	4.13	4.14	4.13	4.11	4.12	4.11
n=1×10⁵	0.46	0.45	0.46	0.47	0.46	0.46	14.83	14.85	14.85	14.82	14.84	14.81
n=2×10⁵	0.46	0.45	0.46	0.47	0.45	0.47	54.34	54.41	54.36	54.34	54.38	54.30
n=5×10⁵	0.47	0.46	0.47	0.47	0.46	0.47	398.28	398.42	398.25	398.22	398.24	398.20
n=1×10⁶	0.47	0.46	0.46	0.47	0.46	0.47	1654.66	1654.81	1654.58	1654.54	1654.78	1654.39
n=5×10⁶	0.45	0.44	0.45	0.45	0.45	0.46	49378.36	49360.86	49361.97	49354.39	49356.79	49353.75

604 **Table B3a.** Comparative metrics of PCHC and PCTABU algorithms regarding the simulated BN
605 built for the data set with 100 nodes and 3 neighbors on average.

	Number of Additions						Number of Deletions					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	29	36.13	28.53	28.60	35.67	28.07	20.13	19.40	20	20.20	19.40	19.87
n=20000	26.40	37.73	26.67	25.13	36.87	25.73	22.53	21.93	22.47	22.73	21.93	22.47
n=50000	31.20	40.87	30.07	29.33	40.13	28.73	17.60	17.07	17.60	17.53	17.07	17.53
n=1×10⁵	32.73	43.87	32.80	30.53	43.67	30.67	14.13	13.93	14.20	14.07	13.93	14.13
n=2×10⁵	42	52.80	41.80	40.33	52.53	40.13	13.53	12.93	13.53	13.60	12.93	13.60
n=5×10⁵	45.93	60.27	46.33	44.20	59.93	44.47	13.47	13.20	13.53	13.40	13.20	13.47
n=1×10⁶	54.07	67.27	54.07	52.47	67.07	51.87	13.60	13.33	13.60	13.53	13.33	13.53
n=5×10⁶	64.80	77.80	64.47	63.33	77.27	63	8	7.73	7.87	8	7.73	7.93
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	18.87	17.87	18.40	19	17.93	18.87	68	73.40	66.93	67.80	73	66.80

n=20000	15.80	16.67	16.07	15.73	16.67	15.93	64.73	76.33	65.20	63.60	75.47	64.13
n=50000	12.27	14	12.07	12	13.73	11.60	61.07	71.93	59.73	58.87	70.93	57.87
n=1×10⁵	13	14.40	12.47	12.67	14.07	12	59.87	72.20	59.47	57.27	71.67	56.80
n=2×10⁵	12.53	14.33	12.20	11.93	14.40	11.60	68.07	80.07	67.53	65.87	79.87	65.33
n=5×10⁵	12.20	14.40	12.27	11.93	14.33	12.07	71.60	87.87	72.13	69.53	87.47	70
n=1×10⁶	12	14.73	12	11.33	14.60	11.47	79.67	95.33	79.67	77.33	95	76.87
n=5×10⁶	12.53	15.07	12.40	12.33	14.93	12	85.33	100.60	84.73	83.67	99.93	82.93

606 **Table B3b.** Comparative metrics of PCHC and PCTABU algorithms regarding the simulated BN
607 built for the data set with 100 nodes and 3 neighbors on average.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.45	0.48	0.46	0.45	0.48	0.46	0.41	0.39	0.41	0.41	0.39	0.41
n=20000	0.51	0.51	0.51	0.51	0.51	0.51	0.49	0.42	0.48	0.50	0.43	0.49
n=50000	0.61	0.59	0.61	0.61	0.60	0.62	0.52	0.45	0.53	0.53	0.46	0.54
n=1×10⁵	0.63	0.61	0.64	0.64	0.62	0.65	0.51	0.44	0.51	0.52	0.44	0.53
n=2×10⁵	0.66	0.64	0.66	0.66	0.64	0.67	0.48	0.42	0.49	0.49	0.42	0.50
n=5×10⁵	0.66	0.64	0.66	0.67	0.64	0.67	0.47	0.40	0.47	0.48	0.40	0.48
n=1×10⁶	0.68	0.64	0.68	0.69	0.65	0.69	0.44	0.38	0.44	0.46	0.38	0.46
n=5×10⁶	0.71	0.68	0.71	0.71	0.68	0.72	0.40	0.34	0.40	0.40	0.34	0.41
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.43	0.42	0.44	0.43	0.42	0.43	0.99	0.99	0.98	0.94	0.95	0.94
n=20000	0.50	0.46	0.50	0.51	0.47	0.50	1.99	1.99	1.98	1.94	1.94	1.94
n=50000	0.56	0.51	0.56	0.57	0.52	0.57	4.63	4.63	4.61	4.57	4.56	4.56
n=1×10⁵	0.56	0.51	0.57	0.57	0.52	0.58	14.92	14.96	14.91	14.86	14.86	14.86
n=2×10⁵	0.55	0.51	0.56	0.56	0.51	0.57	74.83	74.86	74.79	74.73	74.74	74.70
n=5×10⁵	0.55	0.49	0.55	0.56	0.49	0.55	283.69	283.89	283.65	283.59	283.67	283.53
n=1×10⁶	0.53	0.47	0.53	0.55	0.48	0.55	1774.90	1775.28	1774.75	1774.56	1774.84	1774.56
n=5×10⁶	0.51	0.45	0.51	0.51	0.45	0.52	21916.28	21903.54	21893.13	21890	21892.65	21889.70

608 **Table B4a.** Comparative metrics of PCHC and PCTABU algorithms regarding the simulated BN
609 built for the data set with 100 nodes and 5 neighbors on average.

Number of Additions				Number of Deletions			
PCHC		PCTABU		PCHC		PCTABU	

	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	27.00	35.60	28.07	25.53	34.60	26.87	61.27	59.67	61.47	61.20	59.67	61.33
n=20000	32.60	42.93	32.60	30.93	41.40	31.27	65.47	64.00	65.07	65.33	64.00	65.07
n=50000	35.60	44.60	35.40	33.80	44.47	33.80	51.13	49.80	51.00	51.07	49.80	50.87
n=1×10⁵	37.40	46.80	38.27	36.00	46.53	36.87	49.60	48.47	49.40	49.60	48.47	49.33
n=2×10⁵	42.47	53.47	43.07	40.73	53.00	41.67	47.47	46.60	47.67	47.47	46.60	47.60
n=5×10⁵	56.27	66.73	56.73	54.67	66.47	55.27	47.00	46.07	46.80	46.93	46.07	46.80
n=1×10⁶	59.73	71.13	59.80	58.87	70.33	59.47	44.67	44.20	44.67	44.60	44.20	44.60
n=5×10⁶	66.25	78.00	67.25	65.50	78.00	66.00	39.25	38.75	39.50	39.50	38.75	39.50
	Number of Reversals						SHD					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	16.20	16.00	16.47	15.87	15.67	16.47	104.47	111.27	106	102.60	109.93	104.67
n=20000	19.53	18.13	19.00	19.47	17.80	19.07	117.60	125.07	116.67	115.73	123.20	115.40
n=50000	16.07	15.33	15.67	15.87	15.47	15.60	102.80	109.73	102.07	100.73	109.73	100.27
n=1×10⁵	13.20	12.93	13.73	13.07	13.27	13.40	100.20	108.20	101.40	98.67	108.27	99.60
n=2×10⁵	13.73	13.07	13.73	13.53	13.13	13.47	103.67	113.13	104.47	101.73	112.73	102.73
n=5×10⁵	14.80	16.20	14.87	15.13	16.40	14.93	118.07	129.00	118.40	116.73	128.93	117.00
n=1×10⁶	14.53	15.20	14.47	14.60	15.33	14.60	118.93	130.53	118.93	118.07	129.87	118.67
n=5×10⁶	8.25	9.75	8.75	8.25	9.75	8.25	113.75	126.50	115.50	113.25	126.50	113.75

610 **Table B4b.** Comparative metrics of PCHC and PCTABU algorithms regarding the simulated BN
611 built for the data set with 100 nodes and 5 neighbors on average.

	Precision						Recall					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.37	0.39	0.37	0.37	0.39	0.37	0.51	0.48	0.50	0.53	0.49	0.51
n=20000	0.35	0.37	0.36	0.35	0.38	0.36	0.47	0.44	0.48	0.48	0.46	0.48
n=50000	0.45	0.47	0.46	0.46	0.47	0.46	0.52	0.49	0.52	0.53	0.49	0.53
n=1×10⁵	0.50	0.51	0.50	0.50	0.51	0.50	0.55	0.52	0.55	0.56	0.52	0.56
n=2×10⁵	0.52	0.53	0.51	0.52	0.53	0.52	0.54	0.50	0.53	0.55	0.50	0.54
n=5×10⁵	0.52	0.51	0.52	0.52	0.51	0.52	0.48	0.44	0.48	0.48	0.44	0.48
n=1×10⁶	0.54	0.53	0.54	0.54	0.53	0.54	0.48	0.44	0.48	0.48	0.44	0.48
n=5×10⁶	0.63	0.62	0.62	0.62	0.62	0.62	0.52	0.47	0.51	0.52	0.47	0.51
	F-Score						Time					
	PCHC			PCTABU			PCHC			PCTABU		
	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC	BDe	LL	BIC
n=10000	0.43	0.42	0.42	0.43	0.43	0.43	1.84	1.85	1.85	1.79	1.79	1.79
n=20000	0.40	0.41	0.41	0.41	0.41	0.41	4.67	4.69	4.67	4.61	4.63	4.60

n=50000	0.48	0.48	0.49	0.49	0.48	0.49	11.75	11.79	11.77	11.68	11.70	11.70
n=1×10⁵	0.53	0.51	0.52	0.53	0.51	0.53	46.75	46.82	46.79	46.68	46.72	46.68
n=2×10⁵	0.53	0.51	0.52	0.53	0.51	0.53	162.04	162.19	162.10	161.97	162.07	161.96
n=5×10⁵	0.50	0.47	0.50	0.50	0.47	0.50	1452.26	1452.52	1452.32	1452.13	1452.39	1452.14
n=1×10⁶	0.50	0.48	0.51	0.51	0.48	0.51	6385.17	6384.62	6384.04	6383.63	6384.23	6383.68
n=5×10⁶	0.57	0.54	0.56	0.57	0.54	0.57	196955.90	193877	196616.20	193296.90	195307	191645