Learning LWF Chain Graphs: A Markov Blanket Discovery Approach

Mohammad Ali Javidian **Marco Valtorta** **Pooyan Jamshidi**
Computer Science & Engineering Department
University of South Carolina
Columbia, SC 29201

Abstract

This paper provides a graphical characterization of Markov blankets in chain graphs (CGs) under the Lauritzen-Wermuth-Frydenberg (LWF) interpretation. The characterization is different from the well-known one for Bayesian networks and generalizes it. We provide a novel scalable and sound algorithm for Markov blanket discovery in LWF CGs and prove that the Grow-Shrink algorithm, the IAMB algorithm, and its variants are still correct for Markov blanket discovery in LWF CGs under the same assumptions as for Bayesian networks. We provide a sound and scalable constraint-based framework for learning the structure of LWF CGs from faithful causally sufficient data and prove its correctness when the Markov blanket discovery algorithms in this paper are used. Our proposed algorithms compare positively/competitively against the state-of-the-art LCD (**L**earn **C**hain graphs via **D**ecomposition) algorithm, depending on the algorithm that is used for Markov blanket discovery. Our proposed algorithms make a broad range of inference/learning problems computationally tractable and more reliable because they exploit locality.

1 INTRODUCTION

Probabilistic graphical models are now widely accepted as a powerful and mature tool for reasoning and decision making under uncertainty. A *probabilistic graphical model* (PGM) is a compact representation of a joint probability distribution, from which we can obtain marginal and conditional probabilities (Sucar, 2015). In fact, any PGM consists of two main components: (1) a graph that defines the structure of that model; and (2) a joint distribution over random variables of the model. Two types of graphical representations of distributions are commonly used, namely, Bayesian networks (BNs) and Markov networks (MNs). They encompass the properties of factorization and independence, but they differ in the set of independencies they can encode and the factorization of the distribution that they induce.

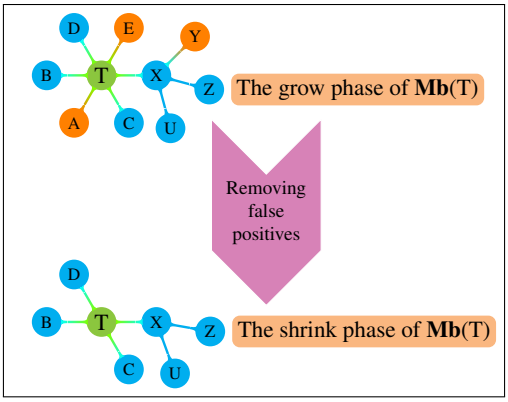

Figure 1: The procedure of Markov blanket recovery in the Grow-Shrink based algorithms.

Currently systems containing both causal and non-causal relationships are mostly modeled with *directed acyclic graphs* (DAGs). Chain graphs (CGs) are a type of mixed graphs, admitting both directed and undirected edges, which contain no partially directed cycles. So, CGs may contain two types of edges, the directed type that corresponds to the causal relationship in DAGs and a second type of edge representing a symmetric relationship (Sonntag and Peña, 2015). *LWF Chain graphs* were introduced by **L**auritzen, **W**ermuth and **F**rydenberg (Frydenberg, 1990; Lauritzen and Wermuth, 1989) as a generalization of graphical models based on undirected graphs and DAGs and widely studied (Lauritzen, 1996; Lauritzen and Richardson, 2002; Drton, 2009; Studený, Roverato, and Štěpánová, 2009; Sonntag and Peña, 2015;

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

Roverato, 2005; Roverato and Rocca, 2006). From the *causality* point of view, in an LWF CG: Directed edges represent *direct causal effect*s. Undirected edges represents causal effects due to *interference* (Shpitser, Tchetgen, and Andrews, 2017; Ogburn, Shpitser, and Lee, 2018; Bhattacharya, Malinsky, and Shpitser, 2019).

One important and challenging aspect of PGMs is the possibility of learning the structure of models directly from sampled data. Five *constraint-based* learning algorithms, which use a statistical analysis to test the presence of a conditional independency, exist for learning LWF CGs: (1) the inductive causation like (IC-like) algorithm (Studený, 1997), (2) the decomposition-based algorithm called LCD (Ma, Xie, and Geng, 2008), (3) the answer set programming (ASP) algorithm (Sonntag et al., 2015), (4) the inclusion optimal (CKES) algorithm (Peña, Sonntag, and Nielsen, 2014), and (5) the local structure learning of chain graphs algorithm with false discovery rate control (Wang, Liu, and Zhu, 2019).

In a DAG G with node set V, each local distribution depends only on a single node $v \in V$ and on its parents (i.e., the nodes $u \neq v$ such that $u \longrightarrow v$, here denoted $pa(v)$). Then the overall joint density is simply $p(x) = \Pi_{v \in V}^{n} p(x_v | x_{pa(v)})$. The key advantage of the decomposition in this equation is to make *local computations* possible for most tasks, using just a few variables at a time regardless of the magnitude of $|V| = n$. In Bayesian networks, the concept that enables us to take advantage of local computation is *Markov blanket*. The Markov blanket (*Markov boundary* in Pearl's terminology) of each node v, defined as the smallest set $\mathbf{Mb}(v)$ of nodes that separates v from all other nodes $V \setminus \{v, \mathbf{Mb}(v)\}$. Markov blankets can be used for variable selection for classification, for causal discovery, and for Bayesian network learning (Tsamardinos et al., 2003).

Markov blanket discovery has attracted a lot of attention in the context of Bayesian network structure learning (see section 2). It is surprising, however, how little attention (if any) it has attracted in the context of learning LWF chain graphs. In this paper, we focus on addressing the problem of Markov blanket discovery for structure learning of LWF chain graphs. For this purpose, we extend the concept of Markov blankets to LWF CGs. We prove that Grow-Shrink Markov Blanket (GSMB) (Margaritis and Thrun, 1999), IAMB, and its variants (Tsamardinos et al., 2003; Yaramakala and Margaritis, 2005) (that are mainly designed for Markov blanket recovery in BNs) are still correct for Markov blanket discovery in LWF CGs under the faithfulness and causal sufficiency assumptions. We propose a new constraint-based Markov blanket recovery algorithm, called MBC-CSP, that is specifically designed for Markov blanket discovery in LWF CGs.

Since constraint-based learning algorithms are sensitive to *error propagation* (Triantafillou, Tsamardinos, and Roumpelaki, 2014), and an erroneous identification of an edge can propagate through the network and lead to erroneous edge identifications or conflicting orientations *even in seemingly unrelated parts of the network*, the learned chain graph model will be unreliable. In order to address the problem of reliable structure learning, we present a generic approach (i.e., the algorithm is independent of any particular search strategy for Markov blanket discovery) based on Markov blanket recovery to learn the structure of LWF CGs from a faithful data. This algorithm first learns the Markov blanket of each node. This preliminary step greatly simplifies the identification of neighbours. This in turn results in a significant reduction in the number of conditional independence tests, and therefore of the overall computational complexity of the learning algorithm. In order to show the effectiveness of this approach, the resulting algorithms are contrasted against LCD on simulated data. We report experiments showing that our proposed generic algorithm (via 6 different instantiations) provides competitive/better performance against the LCD algorithm in our Gaussian experimental settings, depending on the approach that is used for Markov blanket discovery. Our proposed approach has an advantage over LCD because local structural learning in the form of Markov blanket is a theoretically well-motivated and empirically robust learning framework that can serve as a powerful tool in classification and causal discovery (Aliferis et al., 2010). We also note that Markov blankets are useful in their own right, for example in sensor validation and fault analysis (Ibargüengoytla, Sucar, and Vadera, 1996). Code for reproducing our results and its corresponding user manual is available at `https://github.com/majavid/MbLWF2020`. Our main theoretical and empirical contributions are as follows:

(1) We extend the concept of Markov blankets to LWF CGs and we prove what variables make up the Markov blanket of a target variable in an LWF CG (Section 4).
(2) We theoretically prove that the Grow-Shrink, IAMB algorithm and its variants are still sound for Markov blanket discovery in LWF chain graphs under the faithfulness and causal sufficiency assumptions (Section 4).
(3) We present a new algorithm, called MBC-CSP, for learning Markov blankets in LWF chain graphs, and we prove its correctness theoretically (Section 4).
(4) We propose a generic algorithm for structure learning of LWF chain graphs based on the proposed Markov blanket recovery algorithms in Section 4, and we prove its correctness theoretically (Section 5).
(5) We evaluate the performance of 6 instantiations of

the proposed generic algorithm with 6 different Markov blanket recovery algorithms on synthetic Gaussian data, and we show the competitive performance of our method against the LCD algorithm (Section 6).

2 RELATED WORK

Markov Blanket Recovery for Bayesian Networks with Causal Sufficiency Assumption. Margaritis and Thrun (1999) presented the first provably correct algorithm, called Grow-Shrink Markov Blanket (GSMB), that discovers the Markov blanket of a variable from a faithful data under the causal sufficiency assumption. Variants of GSMB were proposed to improve speed and reliability such as the Incremental Association Markov Blanket (IAMB) and its variants (Tsamardinos et al., 2003), Fast-IAMB (Yaramakala and Margaritis, 2005), and IAMB with false discovery rate control (IAMB-FDR) (Peña, 2008). Since in discrete data the sample size required for high-confidence statistical tests of conditional independence in GSMB and IAMB algorithms grows exponentially in the size of the Markov blanket, several sample-efficient algorithms e.g., HITON-MB (Aliferis et al., 2010) and MaxMin Markov Blanket (MMMB) (Tsamardinos et al., 2006) were proposed to overcome the data inefficiency of GSMB and IAMB algorithms. One can find alternative computational methods for Markov blanket discovery that were developed in the past two decades in (Peña, 2007; Liu and Liu, 2016; Ling et al., 2019), among others.

Markov Blanket Recovery without Causal Sufficiency Assumption. Gao and Ji (Gao and Ji, 2016) proposed the latent Markov blanket learning with constrained structure EM algorithm (LMB-CSEM) to discover the Markov blankets in BNs in the presence of unmeasured confounders. However, LMB-CSEM was proposed to find the Markov blankets in a DAG and provides no theoretical guarantees for finding all possible unmeasured confounders in the Markov blanket of the target variable. Recently, Yu et. al. (Yu et al., 2018) proposed a new algorithm, called M3B, to mine Markov blankets in BNs in the presence of unmeasured confounders.

In this paper, we extend the concept of Markov blankets to LWF CGs, which is different from Markov blankets defined in DAGs under the causal sufficiency assumption and also is different from Markov blankets defined in maximal ancestral graphs without assuming causal sufficiency. So, we need new algorithms that are specifically designed for Markov blanket discovery in LWF CGs.

3 DEFINITIONS AND CONCEPTS

Below, we briefly list some of the central concepts used in this paper.

A *route* ω in G is a sequence of nodes (vertices) $v_1, v_2, \ldots, v_n, n \geq 1$, such that $\{v_i, v_{i+1}\}$ is an edge in G for every $1 \leq i < n$. A *section* of a route is a maximal (w.r.t. set inclusion) non-empty set of nodes $v_i \cdots v_j$ s.t. the route ω contains the subroute $v_i - \cdots - v_j$. It is called a *collider section* if $v_{i-1} \rightarrow v_i - \cdots - v_j \leftarrow v_{j+1}$ is a subroute in ω. For any other configuration the section is a non-collider section. A *path* is a route containing only distinct nodes. A *partially directed path* from v_1 to v_n in a graph G is a sequence of n distinct vertices $v_1, v_2, \ldots, v_n (n \geq 2)$, such that
(a) $\forall i (1 \leq i \leq n)$ either $v_i - v_{i+1}$ or $v_i \rightarrow v_{i+1}$, and
(b) $\exists j (1 \leq j \leq n)$ such that $v_j \rightarrow v_{j+1}$.
A partially directed path with $n \geq 3$ and $v_n \equiv v_1$ is called a *partially directed cycle*. If there is a partially directed path from a to b but not b to a, we say that a is an *ancestor* of b. The set of ancestors of b is denoted by $an(b)$, and we generalize the definition to a set of nodes in the obvious way.

Formally, we define the set of parents, children, neighbors, and spouses of a variable (node) in an LWF CG $G = (V, E)$ as follows, respectively: $pa(v) = \{u \in V | u \rightarrow v \in E\}$, $ch(v) = \{u \in V | v \rightarrow u \in E\}$, $ne(v) = \{u \in V | v - u \in E\}$, $sp(v) = \{u \in V | \exists w \in V$ s.t. $u \rightarrow w \leftarrow v$ in $G\}$. The boundary $bd(A)$ of a subset A of vertices is the set of vertices in $V \setminus A$ that are parents or neighbors to vertices in A. The closure of A is $cl(A) = bd(A) \cup A$. If $bd(a) \subseteq A$, for all $a \in A$ we say that A is an *ancestral set*. The smallest ancestral set containing A is denoted by $An(A)$.

An *LWF CG* is a graph in which there are no partially directed cycles. The chain components $\mathcal{T}$ of a CG are the connected components of the undirected graph obtained by removing all directed edges from the CG. A *minimal complex* (or simply a complex) in a CG is an induced subgraph of the form $a \rightarrow v_1 - \cdots\cdots - v_r \leftarrow b$. We say that a is a *complex-spouse* of b and vice versa, and that $csp(a) = \{b \in V | \exists$ a minimal complex of form $a \rightarrow x - \cdots - y \leftarrow b\}$. The *skeleton* of an LWF CG G is obtained from G by changing all directed edges of G into undirected edges. For a CG G we define its *moral graph* G^m as the undirected graph with the same vertex set but with α and β adjacent in G^m if and only if either $\alpha \rightarrow \beta$, $\beta \rightarrow \alpha$, $\alpha - \beta$, or if $\alpha \in csp(\beta)$.

Global Markov property for LWF CGs: For any triple (A, B, S) of disjoint subsets of V such that S separates A from B in $(G_{An(A \cup B \cup S)})^m$, the moral graph of the smallest ancestral set containing $A \cup B \cup S$, indicated

as $A \perp\!\!\!\perp_c B|S$ (read: S *c-separates* A from B in the CG G), we have $A \perp\!\!\!\perp_p B|S$, i.e., A is independent of B given S. In words, if S c-separates A from B in the CG G, then A and B are independent given S. An equivalent path-wise c-separation criterion, which generalizes the d-separation criterion for DAGs, was introduced in (Studený, 1998). A route ω is *active* with respect to a set $S \subseteq V$ if (i) every collider section of ω contains a node of S or $an(S)$, and (ii) every node in a non-collider section on the route is not in S. A route which is not active with respect to S is *intercepted* (blocked) by S. If G is an LWF CG then X and Y are c-separated given S iff there exists no active route between X and Y.

We say that two LWF chain graphs are *Markov equivalent* if they induce the same conditional independence restrictions. Two chain graphs are Markov equivalent if and only if they have the same skeletons and the same minimal complexes (Frydenberg, 1990). Every class of Markov equivalent CGs has a unique CG, called the *largest CG*, with the greatest number of undirected edges (Frydenberg, 1990).

The *Markov condition* is said to hold for a DAG $G = (V, E)$ and a probability distribution $P(V)$ if every variable T is statistically independent of its graphical non-descendants (the set of vertices for which there is no directed path from T) conditional on its graphical parents in P. Pairs $\langle G, P\rangle$ that satisfy the Markov condition satisfy the following implication: $\forall X, Y \in V, \forall Z \subseteq V \setminus \{X, Y\} : (X \perp\!\!\!\perp_d Y|Z \Longrightarrow X \perp\!\!\!\perp_p Y|Z)$. The *faithfulness condition* states that the only conditional independencies to hold are those specified by the Markov condition, formally: $\forall X, Y \in V, \forall Z \subseteq V \setminus \{X, Y\} : (X \not\perp\!\!\!\perp_d Y|Z \Longrightarrow X \not\perp\!\!\!\perp_p Y|Z)$.

Let a Bayesian network $G = (V, E, P)$ be given. Then, V is a set of random variables, (V, E) is a DAG, and P is a joint probability distribution over V. Let $T \in V$. Then the *Markov blanket* $\mathbf{Mb}(T)$ is the set of all parents of T, children of T, and spouses of T. Formally, $\mathbf{Mb}(T) = pa(T) \cup ch(T) \cup sp(T)$.

4 MARKOV BLANKET DISCOVERY IN LWF CHAIN GRAPHS

Let $G = (V, E, P)$ be an LWF chain graph model. Then, V is a set of random variables, (V, E) is an LWF chain graph, and P is a joint probability distribution over V. Let $T \in V$. Then the *Markov blanket* $\mathbf{Mb}(T)$ is the set of all parents of T, children of T, neighbors of T, and complex-spouses of T. Formally, $\mathbf{Mb}(T) = bd(T) \cup ch(T) \cup csp(T)$. We first show that the Markov blanket of the target variable T in an LWF CG probabilistically shields T from the rest of the variables. Under the faithfulness assumption, the Markov blanket is the smallest set with this property. Then, we propose a novel algorithm, called MBC-CSP, that is specifically designed for Markov blanket discovery in LWF CGs. In addition, we prove that GSMB, IAMB and its variants, and MBC-CSP are sound for Markov blanket discovery in LWF CGs under the faithfulness and causal sufficiency assumptions.

Theorem 1 *Let $G = (V, E, P)$ be an LWF chain graph model. Then, $T \perp\!\!\!\perp_p V \setminus \{T, \boldsymbol{Mb}(T)\}|\boldsymbol{Mb}(T)$.*

Proof It is enough to show that for any $A \in V \setminus \{T, \mathbf{Mb}(T)\}$, $T \perp\!\!\!\perp_c A|\mathbf{Mb}(T)$. For this purpose, we prove that any route between A and T in G is blocked by $\mathbf{Mb}(T)$. In the following cases ($A - \!\!* B$, where means $A - B$ or $A \rightarrow B$ and $A *\!\!-\!\!* B$ means $A - B$, $A \rightarrow B$, or $A \leftarrow B$), we assume without loss of generality that T cannot appear between A and B. (If T appears between A and B, the argument for the appropriate case can be applied inductively.)
(1) The route ω between A and T is of the form $A *\!\!-\!\!* \cdots *\!\!-\!\!* B \rightarrow T$. Clearly, B blocks the route ω.
(2) The route ω between A and T is of the form $A *\!\!-\!\!* \cdots *\!\!-\!\!* B - T$. Clearly, B blocks the route ω.
(3) The route ω between A and T is of the form $A *\!\!-\!\!* \cdots *\!\!-\!\!* C *\!\!-\!\!* B \leftarrow T$. We have the following subcases:
(3i) The route ω between A and T is of the form $A *\!\!-\!\!* \cdots *\!\!-\!\!* C \leftarrow B \leftarrow T$. Clearly, B blocks the route ω.
(3ii) The route ω between A and T is of the form $A *\!\!-\!\!* \cdots *\!\!-\!\!* C - B \leftarrow T$. If B is *not* a node on a collider section of ω, B blocks the route ω. However, If B is a node on a collider section of ω, there are nodes D and E ($\neq A, T$) s.t. the route ω has the form of $A *\!\!-\!\!* \cdots *\!\!-\!\!* E \rightarrow D - \cdots - C - B \leftarrow T$. $E \in csp(T)$ blocks the route ω.
(3iii) The route ω between A and T is of the form $A *\!\!-\!\!* \cdots *\!\!-\!\!* C \rightarrow B \leftarrow T$. $C \in sp(T)$ blocks the route ω.
From the global Markov property it follows that every c-separation relation in G implies conditional independence in every joint probability distribution P that satisfies the global Markov property for G. Thus, we have $T \perp\!\!\!\perp_p V \setminus \{T, \mathbf{Mb}(T)\}|\mathbf{Mb}(T)$. ■

Example 1 *Suppose G is the LWF CG in Figure 2). $\boldsymbol{Mb}(T) = \{C, F, G, H, K, L\}$, because $pa(T) = \{C, G\}, ch(T) = \{K\}, ne(T) = \{F\}, csp(T) = \{L, H\}$. Note that if only T's adjacents are instantiated, then T is not c-separated from L and H in G.*

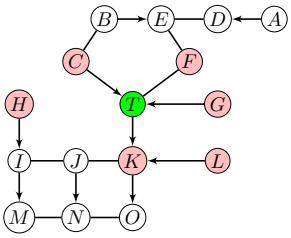

Figure 2: The LWF CG G. The Markov blanket of the target node T is $\mathbf{Mb}(T) = \{C, F, G, H, K, L\}$.

4.1 The MBC-CSP Algorithm for Markov Blanket Discovery in LWF CGs

The MBC-CSP algorithm is structurally similar to the standard Markov blanket discovery algorithms and follows the same two-phase *grow-shrink* structure as shown in the Figure 1. An estimate of the $\mathbf{Mb}(T)$ is kept in the set *CMB*. In the grow phase all variables that belong in $\mathbf{Mb}(T)$ and possibly more (*false positives*) enter *CMB* while in the shrink phase the false positives are identified and removed so that $CMB = \mathbf{Mb}(T)$ in the end.

In the grow phase, MBC-CSP first recovers $adj(T) := pa(T) \cup ch(T) \cup ne(T)$, i.e., the variables adjacent to T. This step is similar to AdjV algorithm in (Yu et al., 2018). Then it discovers complex-spouses of T denoting by $csp(T)$. In the shrink phase, MBC-CSP removes one-by-one the elements of *CMB* that do not belong to the $\mathbf{Mb}(T)$ by testing whether a feature X from *CMB* is independent of T given the remaining *CMB*.

MBC-CSP Description: In Algorithm 1, $adj(T)$ stores the variables adjacent to T, S is the conditioning set, $cor(V_i, T)$ denotes the value of the correlation between V_i and T, $\#adj(T)$ is the number of variables in $adj(T)$, and $\mathbf{Sepset}(T, V_i)$ means the separation set for V_i with respect to T, i.e., the conditioning set that makes T and V_i conditionally independent. From line 1 to 8 of Algorithm 1, MBC-CSP removes the variables that are marginally independent of T and then sorts the remaining variables in an ascending order of their correlations with T. The obtained $adj(T)$ at the end of line 8 may include some false positives. In order to remove false positives from $adj(T)$, we select the variable with the smallest correlation with T, because a variable with a weak correlation with T may have a higher probability to be removed from $adj(T)$ as a false positive than a variable with a strong correlation with T. In this way we speed up the procedure of false positives removal. This procedure begins with a conditioning set of size 1 and then increases the size of the conditioning set one-by-one iteratively until its size is bigger than the size of the current set $adj(T)$. At each iteration, if a variable is found to be independent of T, the variable is removed from the

Algorithm 1: MBC-CSP: An algorithm for Markov blanket discovery in LWF CGs

Input: a data set with variable set V, target variable T, and significance level α.
Output: $\mathbf{Mb}(T)$.

/* **Phase 1: Grow (Forward)** */
/* **step 1:** $adj(T) := pa(T) \cup ch(T) \cup ne(T)$, the set of variables adjacent to T. */

1 **for** $(V_i \in V \setminus \{T\})$ **do**
2 $p_{V_i} = pvalue(T \perp\!\!\!\perp V_i | \varnothing)$;
3 **if** $(p_{V_i} > \alpha)$ **then**
4 $\mathbf{Sepset}(T, V_i) = \varnothing$; /* T is marginally independent of V_i. */
5 **else**
6 Add V_i to $adj(T)$;
7 **end**
8 **end**
9 Sort $adj(T)$ in increasing value of $cor(V_i, T)$;
10 Set $k = 1, \#adj = |adj(T)|$;
11 **while** $(k \leq \#adj)$ **do**
12 **for** $(V_j \in adj(T))$ **do**
13 **if** $(\exists S \subseteq adj(T) \setminus \{V_j\} s.t. T \perp\!\!\!\perp V_j | S \text{ and } |S| = k)$ **then**
14 $adj(T) = adj(T) \setminus V_j$;
15 $\mathbf{Sepset}(T, V_j) = S$;
16 **end**
17 **end**
18 $k = k + 1, \#adj = |adj(T)|$;
19 **end**
20 /* **step 2:** $csp(T)$, complex-spouses of T. */
21 **for** $V_i \in adj(T)$ **do**
22 **for** $V_j \in V \setminus \{adj(T), T\}$ **do**
23 $pval_1 = pvalue(T \perp\!\!\!\perp V_j | \mathbf{Sepset}(T, V_j))$;
24 $pval_2 = pvalue(T \perp\!\!\!\perp V_j | (\mathbf{Sepset}(T, V_j)) \cup \{V_i\})$;
25 **if** $pval_1 > \alpha$ *and* $pval_2 < \alpha$ **then**
26 Add V_j to $csp(T)$;
27 **end**
28 **end**
29 **end**
30 $CMB = adj(T) \cup csp(T)$;
/* **Phase 2: Shrink (Backward)** */
31 *continue* = *TRUE*;
32 **if** $(|CMB| = 0)$ **then**
33 *continue* = *FALSE*;
34 **end**
35 **while** *(continue)* **do**
36 $P_Y = \underset{Y \in CMB}{pvalue}(T \perp\!\!\!\perp Y | CMB \setminus \{Y\})$;
37 $p.val.max = \max_{Y \in CMB} P_Y$;
38 $Candidas = \{Y \in CMB | P_Y = p.val.max\}$;
39 **if** $(p.val.max > \alpha)$ **then**
 /* i.e., $T \perp\!\!\!\perp Y | CMB \setminus \{Y\}$ */
40 $CMB = CMB \setminus Candidas[1]$;
 /* $Candidas[1]$ means the first element of $Candidas$. */
41 **else**
42 *continue* = *FALSE*;
43 **end**
44 **end**
45 return *CMB*;

current $adj(T)$ (line 9 to 19 of Algorithm 1). Now, we need to add the complex-spouses of T to the obtained set at the end of line 19. For this purpose, lines 21-29 find the set of $csp(T)$ by checking the following conditions for each $V_j \in V \setminus \{adj(T), T\}$: $T \perp\!\!\!\perp V_j | \mathbf{Sepset}(T, V_j)$ and $T \not\perp\!\!\!\perp V_j | (\mathbf{Sepset}(T, V_j)) \cup \{V_i\})$. According to the global Markov property for LWF CGs, these two conditions together guarantee that $V_j \in csp(T)$. At the end of line 30, we obtain a candidate set for the Markov blanket of T that may contain some false positives. The phase 2 of Algorithm 1 i.e., lines 35-44, uses the same idea of the shrinking phase of Markov blanket discovery algorithm IAMB (Tsamardinos et al., 2003) for the output of phase 1 to reduce the number of false positives in the output of the algorithm. For this purpose, we remove one-by-one the variables that do not belong to the $\mathbf{Mb}(T)$ by testing whether a variable Y from *CMB* is independent of T given the remaining variables in *CMB*.

Remark 2 *For the adjacency recovery phase of Algorithm 1 (line 1-19), one can use the HITON-PC or MMPC (Aliferis et al., 2010) algorithms, especially in cases where a sample-efficient algorithm is needed.*

Theorem 3 *Given the Markov assumption, the faithfulness assumption, a graphical model represented by an LWF CG, and i.i.d. sampling, in the large sample limit, the Markov blanket recovery algorithms GS (Margaritis and Thrun, 1999), IAMB (Tsamardinos et al., 2003), MMBC-CSP (Algorithm 1), fastIAMB (Yaramakala and Margaritis, 2005), Interleaved Incremental Association (interIAMB) (Tsamardinos et al., 2003), and fdrIAMB (Peña, 2008) correctly identify all Markov blankets for each variable. (Note that* Causal Sufficiency *is assumed i.e., all causes of more than one variable are observed.)*

Proof [Sketch of proof] If a variable belongs to $\mathbf{Mb}(T)$, then it will be admitted in the first step (Grow phase) at some point, since it will be dependent on T given the candidate set of $\mathbf{Mb}(T)$. This holds because of the faithfulness and because the set $\mathbf{Mb}(T)$ is the minimal set with that property. If $X \notin \mathbf{Mb}(T)$, then conditioned on $\mathbf{Mb}(T) \setminus \{X\}$, it will be independent of T and thus will be removed from the candidate set of $\mathbf{Mb}(T)$ in the second phase (Shrink phase) because the Markov condition entails that independencies in the distribution are represented in the graph. Since the faithfulness condition entails dependencies in the distribution from the graph, we never remove any variable X from the candidate set of $\mathbf{Mb}(T)$ if $X \in \mathbf{Mb}(T)$. Using this argument inductively we will end up with the $\mathbf{Mb}(T)$. ■

Algorithm 2: MbLWF: An algorithm for learning LWF CGs via Markov blanket discovery

Input: a set V of nodes and a probability distribution p faithful to an unknown LWF chain graph $G = (V, E)$.
Output: The pattern of G.

/* **Phase 1: Learning Markov blankets** */
1 For each variable $X_i \in V$, learn its Markov blanket $\mathbf{Mb}(X_i)$;
2 Check whether the Markov blankets are symmetric, e.g., $X_i \in \mathbf{Mb}(X_j) \leftrightarrow X_j \in \mathbf{Mb}(X_i)$. Assume that nodes for whom symmetry does not hold are false positives and drop them from each other's Markov blankets;
3 Set $\mathbf{Sepset}(X_i, X_j) = \mathbf{Sepset}(X_j, X_i)$ to the smallest of $\mathbf{Mb}(X_i)$ and $\mathbf{Mb}(X_j)$ if $X_i \notin \mathbf{Mb}(X_j)$ and $X_j \notin \mathbf{Mb}(X_i)$;
/* **Phase 2: Skeleton Recovery** */
4 Construct the undirected graph $H = (V, E)$, where $E = \{X_i - X_j | X_j \in \mathbf{Mb}(X_i)$ and $X_i \in \mathbf{Mb}(X_j)\}$;
5 **for** $i \leftarrow 0$ **to** $|V_H| - 2$ **do**
6 **while** *possible* **do**
7 Select any ordered pair of nodes u and v in H such that $u \in ad_H(v)$ and $|ad_H(u) \setminus v| \geq i$;
 /* $ad_H(x) := \{y \in V | x - y \in E\}$ */
8 **if** *there exists* $S \subseteq (ad_H(u) \setminus v)$ *s.t.* $|S| = i$ *and* $u \perp\!\!\!\perp_p v | S$ *(i.e., u is independent of v given S in the probability distribution p)* **then**
9 Set $S_{uv} = S_{vu} = S$;
10 Remove the edge $u - v$ from H;
11 **end**
12 **end**
13 **end**
/* **Phase 3: Complex Recovery (Ma, Xie, and Geng, 2008)** */
14 Initialize $H^* = H$;
15 **for** *each vertex pair* $\{u, v\}$ *s.t. u and v are not adjacent in H* **do**
16 **for** *each* $u - w$ *in* H^* **do**
17 **if** $u \not\perp\!\!\!\perp_p v | (S_{uv} \cup \{w\})$ **then**
18 Orient $u - w$ as $u \longrightarrow w$ in H^*;
19 **end**
20 **end**
21 **end**
22 Take the pattern of H^*;

5 LEARNING LWF CGs VIA MARKOV BLANKETS

Any sound algorithm for learning Markov blankets of LWF CGs can be employed and extended to a full LWF CG learning algorithm, as originally suggested in (Margaritis and Thrun, 1999) for Grow-Shrink Markov blanket algorithm (for Bayesian networks). Thanks to the proposed Markov blanket discovery algorithms listed in Theorem 3, we can now present a generic algorithm for learning LWF CGs. Algorithm 2 lists pseudocode for the three main phases of this approach.

Phase 1: Learning Markov blankets: This phase consists of learning the Markov blanket of each variable with feature selection to reduce the number of candidate structures early on. Any algorithm in Theorem 3 can be

plugged in Step 1. Once all Markov blankets have been learned, they are checked for consistency (Step 2) using their symmetry; by definition $X_i \in \mathbf{Mb}(X_j) \leftrightarrow X_j \in \mathbf{Mb}(X_i)$. Asymmetries are corrected by treating them as false positives and removing those variables from each others Markov blankets. At the end of this phase, separator sets of X and Y set to the smallest of $\mathbf{Mb}(X)$ and $\mathbf{Mb}(Y)$ if $X \notin \mathbf{Mb}(Y)$ and $Y \notin \mathbf{Mb}(X)$.

Phase 2: Skeleton Recovery: First, we construct the moral graph of the LWF CG G that is an undirected graph in which each node of the original G is now connected to its Markov blanket (line 4 of Algorithm 2). Lines 5-13 learn the *skeleton* of the LWF CG by removing the spurious edges. In fact, we remove the added undirected edge(s) between each variable T and its complex-spouses due to the fact that $csp(T) \subseteq \mathbf{Mb}(T)$. Separation sets are updated correspondingly.

Phase 3: Complex Recovery: We use an approach similar to the proposed algorithm by (Ma, Xie, and Geng, 2008) for complex recovery. To get the pattern of H^* in line 22, at each step, we consider a pair of candidate complex arrows $u_1 \to w_1$ and $u_2 \to w_2$ with $u_1 \neq u_2$, then we check whether there is an undirected path from w_1 to w_2 such that none of its intermediate vertices is adjacent to either u_1 or u_2. If there exists such a path, then $u_1 \to w_1$ and $u_2 \to w_2$ are labeled (as complex arrows). We repeat this procedure until all possible candidate pairs are examined. The pattern is then obtained by removing directions of all unlabeled as complex arrows in H^* (Ma, Xie, and Geng, 2008). Note that one can use three basic rules, namely the *transitivity rule*, the *necessity rule*, and the *double-cycle rule*, for changing the obtained pattern in the previous phase into the corresponding largest CG (see Studený (1997) for details).

Computational Complexity Analysis of Algorithm 2 Assume that the "learning Markov blankets" phase uses the grow-shrink (GSMB) approach and $n = |V|$, $m = |E|$, where $G = (V, E)$ is the true LWF CG. Since the Markov blanket algorithm involves $O(n)$ conditional independence (CI) tests, Phase 1 (learning Markov blankets) involves $O(n^2)$ tests. If $b = max_X|\mathbf{Mb}(X)|$, the skeleton recovery (line 5-13) does $O(nb2^b)$ CI tests. In the worst case, i.e. when $b = O(n)$ and $m = O(n^2)$ i.e. the original graph is dense, the total complexity for these 2 phases becomes $O(n^2 + nb2^b)$ or $O(n^2 2^n)$. Under the assumption that b is bounded by a constant (the sparseness assumption), the complexity of Phase 1 and 2 together is $O(n^2)$ in the number of CI tests. As claimed in (Ma, Xie, and Geng, 2008), the total complexity of Phase 3 (complex recovery, lines 14-22) is $O(mn)$ in the number of CI tests. The total number of CI tests for the entire algorithm is therefore $O(n^2 + nb2^b + mn)$. Under the assumption that b is bounded by a constant, this algorithm is $O(n^2 + mn)$ in the number of CI tests.

6 Experimental Evaluation

We performed a large set of experiments on simulated data for contrasting: (1) our proposed Markov blanket discovery algorithm, MBC-CSP, against GS, IAMB, fastIAMB, interIAMB, and fdrIAMB for Markov blanket recovery only, due to their important role in causal discovery and classification; and (2) our proposed structure learning algorithms (GSLWF, IAMBLWF, interIAMBLWF, fastIAMBLWF, fdrIAMBLWF, and MBCCSPLWF) against the state-of-the-art algorithm LCD for LWF CG recovery. We implemented all algorithms in R by extending code from the bnlearn (Scutari, 2010) and pcalg (Kalisch et al., 2012) packages to LWF CGs. We run our algorithms and the LCD algorithm on randomly generated LWF CGs and we compare the results and report summary error measures.

Experimental Settings: Let $N = 2$ or 3 denote the average degree of edges (including undirected, pointing out, and pointing in) for each vertex. We generated random LWF CGs with 30, 40, or 50 variables and $N = 2$ or 3, as described in (Ma, Xie, and Geng, 2008) (see Appendix B for details). Then, we generated Gaussian distributions of size 200 and 2000 on the resulting LWF CGs via the rnorm.cg function from the LCD R package, respectively. For each sample, two different significance levels ($\alpha = 0.05, 0.005$) are used to perform the hypothesis tests. The *null hypothesis* H_0 is "two variables u and v are conditionally independent given a set C of variables" and alternative H_1 is that H_0 may not hold. We then compare the results to access the influence of the significance testing level on the performance of our algorithms.

Metrics for Evaluation: We evaluate the performance of the proposed algorithms in terms of the six measurements that are commonly used (Colombo and Maathuis, 2014; Tsamardinos et al., 2006) for constraint-based algorithms: (a) the true positive rate (TPR) (also known as recall), (b) the false positive rate (FPR), (c) the true discovery rate (TDR) (also known as precision), (d) accuracy (ACC) for the skeleton, (e) the Structural Hamming Distance (SHD), and (f) run-time. In principle, large values of TPR, TDR, and ACC, and small values of FPR and SHD indicate good performance.

6.1 Results and their Implications

Our experimental results for LWF CGs with 50 variables are partially (only for a few configurations of parameters) shown in Figures 3 and 4. The other results are in Appendix B. We did not test whether the faithfulness as-

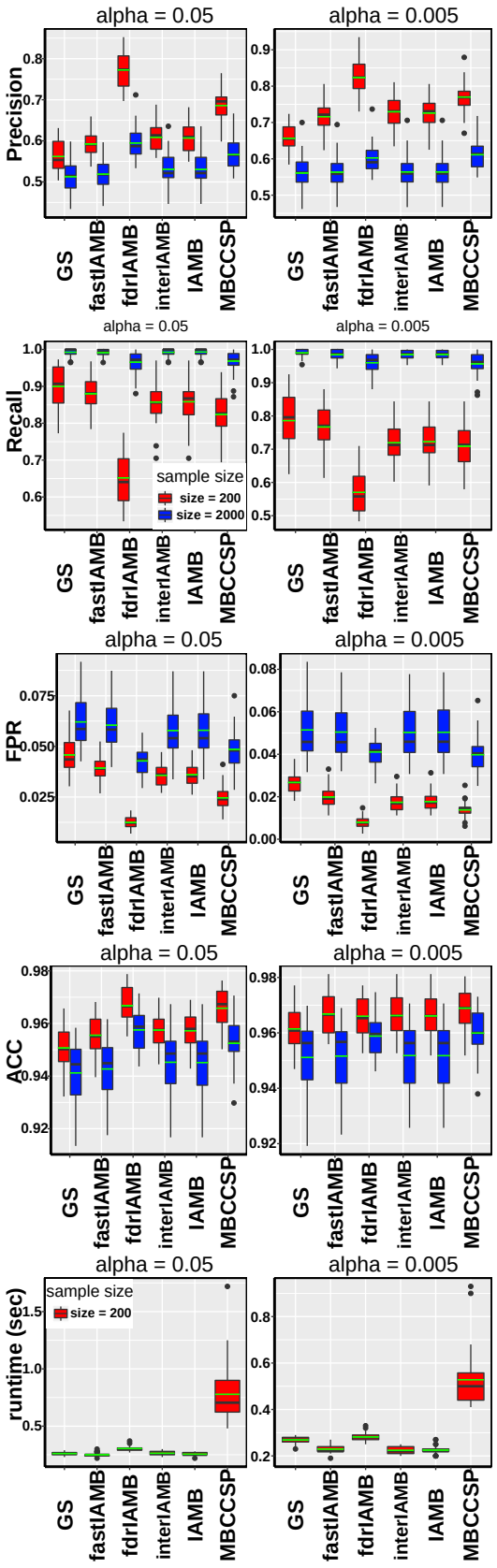

Figure 3: Performance of Markov blanket recovery algorithms for randomly generated Gaussian chain graph models: over 30 repetitions with 50 variables correspond to N = 3. The green line in a box indicates the mean of that group.

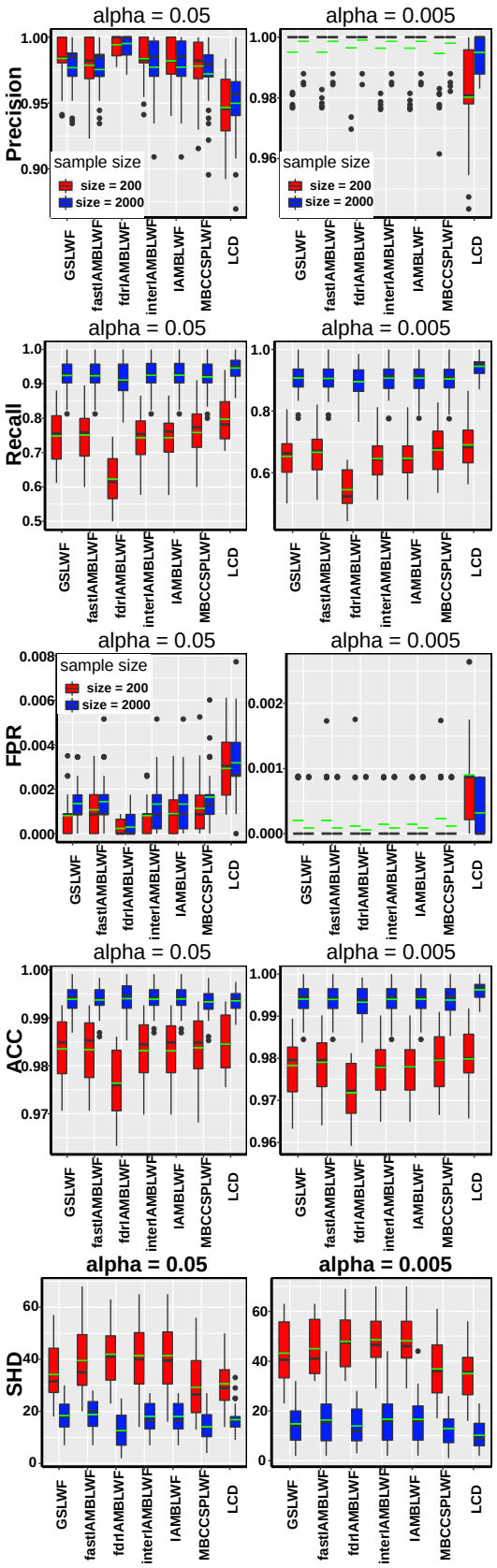

Figure 4: Performance of LCD and MbLWF algorithms for randomly generated Gaussian chain graph models: over 30 repetitions with 50 variables correspond to N = 3. The green line in a box indicates the mean of that group.

sumption holds for any of the networks, thus the results are indicative of the performance of the algorithms on arbitrary LWF CGs.

Some highlights for Markov blanket discovery: (1) As shown in our experimental results, our proposed Markov blanket discovery algorithm, MBC-CSP, is as good as or even (slightly) better than others in many settings. (2) As expected, the recall of all algorithms increases with an increase in sample size. Surprisingly, however, the other error measures worsen with an increase in sample size. A possible explanation could be that the correlation test is too aggressive and rejects variables that are in fact related in the ground truth model. (3) The significance level (p-value or α parameter) has a notable impact on the performance of algorithms. Except for precision, the lower the significance level, the better the performance. (4) The fdrIAMB algorithm has the best precision, FPR, and ACC in small sample size settings, which is consistent with previously reported results (Peña, 2008). This comes at the expense, however, of much worse recall.

Some highlights for LWF CGs recovery: (1) As shown in our experimental results, our proposed Markov blanket based algorithm, MbLWF, is as good as or even (slightly) better than LCD in many settings. The reason is that both LCD and MbLWF algorithms take advantage of local computations that make them equally robust against the choice of learning parameters. (2) While our Markov blanket based algorithms have better precision and FPR, the LCD algorithm enjoys (slightly) better recall. The reason for this may be that the faithfulness assumption makes the LCD algorithm search for a CG that represents all the independencies that are detected in the sample set. However, such a CG may also represent many other independencies. Therefore, the LCD algorithm trades precision for recall. In other words, it seems that the faithfulness assumption makes the LCD algorithm overconfident and aggressive, whereas under this assumption MbLWF algorithms are more cautious, conservative, and more importantly more precise than the LCD algorithm. (3) Except for the fdrIAMB algorithm in small sample size, there is no meaningful difference among the performance of algorithms based on ACC. (4) The best SHD belongs to MBC-CSPLWF and LCD in small sample size settings, and to MBC-CSPLWF, fdrIAMB, and LCD in large sample size settings. (5) Constraint-based learning algorithms always have been criticized for their relatively high structural-error rate (Triantafillou, Tsamardinos, and Roumpelaki, 2014). However, as shown in our experimental results, the proposed Markov blanket based approach is, overall, as good as or even better than the state-of the-art algorithm, i.e., LCD. One of the most important implications of this work is that there is much room for improvement to the constraint-based algorithms in general and Markov blanket based learning algorithms in particular, and hopefully this work will inspire other researchers to address this important class of algorithms. (6) Markov blankets of different variables can be learned independently from each other, and later merged and reconciled to produce a coherent LWF CG. This allows the parallel implementations for scaling up the task of learning chain graphs from data containing more than hundreds of variables, which is crucial for big data analysis tools. In fact, our proposed structure learning algorithms can be parallelized following (Scutari, 2017); see supplementary material for a detailed example.

With the use of our generic algorithm (Algorithm 2), the problem of structure learning is reduced to finding an efficient algorithm for Markov blanket discovery in LWF CGs. This greatly simplifies the structure-learning task and makes a wide range of inference/learning problems computationally tractable because they exploit locality. In fact, due to the causal interpretation of LWF CGs (Richardson and Spirtes, 2002; Bhattacharya, Malinsky, and Shpitser, 2019), discovery of Markov blankets in LWF CGs is significant because it can play an important role for estimating causal effects under unit dependence induced by a network represented by a CG model, when there is uncertainty about the network structure.

7 DISCUSSION AND CONCLUSION

An important novelty of local methods in general and Markov blanket recovery algorithms in particular for structure learning is circumventing non-uniform graph connectivity. A chain graph may be non-uniformly dense/sparse. In a global learning framework, if a region is particularly dense, that region cannot be discovered quickly and many errors will result when learning with a small sample. These errors propagate to remote regions in the chain graph including those that are learnable accurately and fast with local methods. In contrast, local methods such as Markov blanket discovery algorithms are fast and accurate in the less dense regions. In addition, when the dataset has tens or hundreds of thousands of variables, applying global discovery algorithms that learn the full chain graph becomes impractical. In those cases, Markov blanket based approaches that take advantage of local computations can be used for learning full LWF CGs. For this purpose, we extended the concept of Markov blankets to LWF CGs and we proposed a new algorithm, called MBC-CSP, for Markov blanket discovery in LWF CGs. We proved that GSMB and IAMB and its variants are still sound for Markov blanket discovery in LWF CGs under the faithfulness and causal sufficiency assumptions. This, in turn, enabled us to ex-

tend these algorithms to a new family of global structure learning algorithms based on Markov blanket discovery. As we have shown for the MBC-CSP algorithm, having an effective strategy for Markov blanket recovery in LWF CGs improves the quality of the learned Markov blankets, and consequently the learned LWF CG.

As noticed by Li and Wang (2009), the choice of which performance parameter to optimize (equivalently, which error parameter to control) depends on the application, so we reported on several performance parameters in our experiments. We plan to address the multiple hypotheses testing problem in the small sample case in future work. An approach based on the theoretical work in (Benjamini and Yekutieli, 2001) that uses explicit control of error rates was attempted and carried out in (Wang, Liu, and Zhu, 2019).

Another interesting direction for future work is answering the following question: Can we relax the faithfulness assumption and develop a correct, scalable, and data efficient algorithm for learning Markov blankets in LWF CGs?

Acknowledgements

This work has been supported by AFRL and DARPA (FA8750-16-2-0042). This work is also partially supported by an ASPIRE grant from the Office of the Vice President for Research at the University of South Carolina. We appreciate the comments, suggestions, and questions from all anonymous reviewers and thank them for the careful reading of our paper.

References

Aliferis, C. F.; Statnikov, A.; Tsamardinos, I.; Mani, S.; and Koutsoukos, X. D. 2010. Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation. *J. Mach. Learn. Res.* 11:171–234.

Benjamini, Y., and Yekutieli, D. 2001. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics* 29(4):1165–1188.

Bhattacharya, R.; Malinsky, D.; and Shpitser, I. 2019. Causal inference under interference and network uncertainty. In *Proceedings of the UAI 2019*.

Colombo, D., and Maathuis, M. H. 2014. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.* 15(1):3741–3782.

Drton, M. 2009. Discrete chain graph models. *Bernoulli* 15(3):736–753.

Frydenberg, M. 1990. The chain graph Markov property. *Scandinavian Journal of Statistics* 17(4):333–353.

Gao, T., and Ji, Q. 2016. Constrained local latent variable discovery. In *Proceedings of IJCAI'16*, 1490–1496.

Ibargüengoytla, P. H.; Sucar, L. E.; and Vadera, S. 1996. A probabilistic model for sensor validation. In *Proceedings of the UAI96*, 332–339.

Javidian, M. A., and Valtorta, M. 2018. Finding minimal separators in LWF chain graphs. In *Proceedings of the PGM 2018*, 193–200.

Kalisch, M.; Mchler, M.; Colombo, D.; Maathuis, M.; and Bühlmann, P. 2012. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software, Articles* 47(11):1–26.

Lauritzen, S., and Richardson, T. 2002. Chain graph models and their causal interpretations. *Journal of the Royal Statistical Society. Series B, Statistical Methodology* 64(3):321–348.

Lauritzen, S., and Wermuth, N. 1989. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics* 17(1):31–57.

Lauritzen, S. 1996. *Graphical Models*. Oxford Science Publications.

Li, J., and Wang, Z. J. 2009. Controlling the false discovery rate of the association/causality structure learned with the PC algorithm. *J. Mach. Learn. Res.* 10:475514.

Ling, Z.; Yu, K.; Wang, H.; Liu, L.; Ding, W.; and Wu, X. 2019. Bamb: A balanced Markov blanket discovery approach to feature selection. *ACM Trans. Intell. Syst. Technol.* 10(5).

Liu, X., and Liu, X. 2016. Swamping and masking in Markov boundary discovery. *Machine Learning* 104(1):25–54.

Ma, Z.; Xie, X.; and Geng, Z. 2008. Structural learning of chain graphs via decomposition. *Journal of Machine Learning Research* 9:2847–2880.

Margaritis, D., and Thrun, S. 1999. Bayesian network induction via local neighborhoods. In *Proceedings of the NIPS'99*, 505–511.

Ogburn, E.; Shpitser, I.; and Lee, Y. 2018. Causal inference, social networks, and chain graphs.

Peña, J. M.; Sonntag, D.; and Nielsen, J. 2014. An inclusion optimal algorithm for chain graph structure learning. *In Proceedings of the AISTATS* 778–786.

Peña, J. M. 2007. Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning* 45(2):211 – 232.

Peña, J. M. 2008. Learning Gaussian graphical models of gene networks with false discovery rate control. In *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, 165–176.

Richardson, T. S., and Spirtes, P. 2002. Ancestral graph Markov models. *The Annals of Statistics* 30(4).

Roverato, A., and Rocca, L. L. 2006. On block ordering of variables in graphical modelling. *Scandinavian Journal of Statistics* 33(1):65–81.

Roverato, A. 2005. A unified approach to the characterization of equivalence classes of DAGs, chain graphs with no flags and chain graphs. *Scandinavian Journal of Statistics* 32(2):295–312.

Scutari, M. 2010. Learning Bayesian networks with the bnlearn R Package. *Journal of Statistical Software* 35(3):1–22.

Scutari, M. 2017. Bayesian network constraint-based structure learning algorithms: Parallel and optimized implementations in the bnlearn R package. *Journal of Statistical Software, Articles* 77(2).

Shpitser, I.; Tchetgen, E. T.; and Andrews, R. 2017. Modeling interference via symmetric treatment decomposition.

Sonntag, D., and Peña, J. M. 2015. Chain graph interpretations and their relations revisited. *International Journal of Approximate Reasoning* 58:39 – 56.

Sonntag, D.; Jãrvisalo, M.; Peña, J. M.; and Hyttinen, A. 2015. Learning optimal chain graphs with answer set programming. In *Proceedings of the UAI31*, 822–831.

Studený, M.; Roverato, A.; and Štěpánová, Š. 2009. Two operations of merging and splitting components in a chain graph. *Kybernetika* 45(2):208–248.

Studený, M. 1997. A recovery algorithm for chain graphs. *International Journal of Approximate Reasoning* 17:265–293.

Studený, M. 1998. Bayesian networks from the point of view of chain graphs. In *Proceedings of the UAI'98*, 496–503.

Sucar, L. E. 2015. *Probabilistic Graphical Models: Principles and Applications*. Springer, London.

Triantafillou, S.; Tsamardinos, I.; and Roumpelaki, A. 2014. Learning neighborhoods of high confidence in constraint-based causal discovery. In van der Gaag, L. C., and Feelders, A. J., eds., *Probabilistic Graphical Models*, 487–502.

Tsamardinos, I.; Aliferis, C.; Statnikov, A.; and Statnikov, E. 2003. Algorithms for large scale Markov blanket discovery. In *In The 16th International FLAIRS Conference, St*, 376–380. AAAI Press.

Tsamardinos, I.; ; Brown, L. E.; and Aliferis, C. F. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning* 65(1).

Wang, J.; Liu, S.; and Zhu, M. 2019. Local structure learning of chain graphs with the false discovery rate control. *Artif. Intell. Rev.* 52(1):293321.

Yaramakala, S., and Margaritis, D. 2005. Speculative Markov blanket discovery for optimal feature selection. In *Proceedings of the ICDM'05*.

Yu, K.; Liu, L.; Li, J.; and Chen, H. 2018. Mining Markov blankets without causal sufficiency. *IEEE Transactions on Neural Networks and Learning Systems* 29(12):6333–6347.

Appendix A: Correctness of Algorithm 2

We prove the correctness of the Algorithm 2 with following lemmas.

Lemma 4 *After line 13 of Algorithm 2, G and H have the same adjacencies.*

Proof Consider any pair of nodes A and B in G. If $A \in ad_G(B)$, then $A \not\perp\!\!\!\perp B|S$ for all $S \subseteq V \setminus (A \cup B)$ by the faithfulness assumption. Consequently, $A \in ad_H(B)$ at all times. On the other hand, if $A \notin ad_G(B)$ (equivalently $B \notin ad_G(A)$), Algorithm 3 (Javidian and Valtorta, 2018) returns a set $Z \subseteq ad_H(A) \setminus B$ (or $Z \subseteq ad_H(B) \setminus A$) such that $A \perp\!\!\!\perp_p B|Z$. This means there exist $0 \leq i \leq |V_H| - 2$ such that the edge $A - B$ is removed from H in line 10. Consequently, $A \notin ad_H(B)$ after line 13.

Algorithm 3: Minimal separation

Input: Two non-adjacent nodes A, B in the LWF chain graph G.

Output: Set Z, that is a minimal separator for A, B.

1 Construct $G_{An(A\cup B)}$;
2 Construct $(G_{An(A\cup B)})^m$;
3 Set Z' to be $ne(A)$ (or $ne(B)$) in $(G_{An(A\cup B)})^m$;
`/* Z' is a separator because, according to the local Markov property of an undirected graph, a vertex is conditionally independent of all other vertices in the graph, given its neighbors (Lauritzen, 1996). */`
4 Starting from A, run BFS. Whenever a node in Z' is met, mark it if it is not already marked, and do not continue along that path. When BFS stops, let Z'' be the set of nodes which are marked. Remove all markings;
5 Starting from B, run BFS. Whenever a node in Z'' is met, mark it if it is not already marked, and do not continue along that path. When BFS stops, let Z be the set of nodes which are marked;
6 **return** Z;

■

Lemma 5 *G and H^* have the same minimal complexes and adjacencies after line 22 of Algorithm 2.*

Proof G and H^* have the same adjacencies by Lemma 4. Now we show that any arrow that belongs to a minimal complex in G is correctly oriented in line 18 of Algorithm 2, in the sense that it is an arrow with the same orientation in G. For this purpose, consider the following two cases:

Case 1: $u \to w \leftarrow v$ is an induced subgraph in G. So, u, v are not adjacent in H (by Lemma 4), $u - w \in H^*$ (by Lemma 4), and $u \not\perp\!\!\!\perp_p v|(S_{uv} \cup \{w\})$ by the faithfulness assumption. So, $u - w$ is oriented as $u \to w$ in H^* in line 15. Obviously, we will not orient it as $w \to u$.

Case 2: $u \to w - \cdots - z \leftarrow v$, where $w \neq z$ is a minimal complex in G. So, u, v are not adjacent in H (by Lemma 4), $u - w \in H^*$ (by Lemma 4), and $u \not\perp\!\!\!\perp_p v|(S_{uv} \cup \{w\})$ by the faithfulness assumption. So, $u - w$ is oriented as $u \to w$ in H^* in line 15. Since $u \in S_{vw}$ and $w \perp\!\!\!\perp_p v|(S_{wv} \cup \{u\})$ by the faithfulness assumption so u, v, and w do not satisfy the conditions and hence we will not orient $u - w$ as $w \to u$.

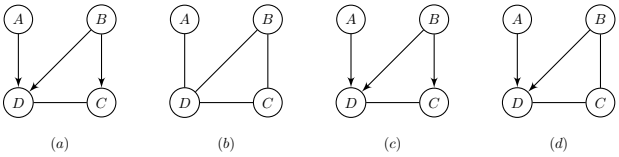

Figure 5: (a) The LWF CG G, (b) the skeleton of G, (c) H^* before executing the line 22 in Algorithm 2, and (d) H^* after executing the line 22 in Algorithm 2.

Consider the chain graph G in Figure 5(a). After applying the skeleton recovery of Algorithm 2, we obtain H, the skeleton of G, in Figure 5(b). In the execution of the complex recovery of Algorithm 2, when we pick A, B in line 15 and C in line 16, we have $A \perp\!\!\!\perp B|\varnothing$, that is, $S_{AB} = \varnothing$, and find that $A \not\perp\!\!\!\perp B|C$. Hence we orient $B - C$ as $B \to C$ in line 18, which is not a complex arrow in G. Note that we do not orient $C - B$ as $C \to B$: the only chance we might do so is when $u = C, v = A$, and $w = B$ in the inner loop of the complex recovery of Algorithm 2, but we have $B \in S_{AC}$ and the condition in line 17 is not satisfied. Hence, the graph we obtain before the last step of complex recovery in Algorithm 2 must be the one given in Figure 5(c), which differs from the recovered pattern in Figure 5(d). This illustrates the necessity of the last step of complex recovery in Algorithm 2. To see how the edge $B \to C$ is removed in the last step of complex recovery in Algorithm 2, we observe that, if we follow the procedure described in the comment after line 22 of Algorithm 2, the only chance that $B \to C$ becomes one of the candidate complex arrow pair is when it is considered together with $A \to D$. However, the only undirected path between C and D is simply $C - D$ with D adjacent to B. Hence $B \to C$ stays unlabeled and will finally get removed in the last step of complex recovery in Algorithm 2.

Consequently, G and H^* have the same minimal complexes and adjacencies after line 22. ■

Appendix B: More Experimental Results

Data Generation Procedure First we explain the way in which the random LWF CGs and random samples are generated. Given a vertex set V, let $p = |V|$ and N denote the average degree of edges (including undirected, pointing out, and pointing in) for each vertex. We generate a random LWF CG on V as follows:

(1) Order the p vertices and initialize a $p \times p$ adjacency matrix A with zeros;

(2) For each element in the lower triangle part of A, set it to be a random number generated from a Bernoulli distribution with probability of occurrence $s = N/(p-1)$;

(3) Symmetrize A according to its lower triangle;

(4) Select an integer k randomly from $\{1, \ldots, p\}$ as the number of chain components;

(5) Split the interval $[1, p]$ into k equal-length subintervals $I_1, \ldots, I_k$ so that the set of variables falling into each subinterval I_m forms a chain component C_m;

(6) Set $A_{ij} = 0$ for any (i, j) pair such that $i \in I_l, j \in I_m$ with $l > m$.

This procedure yields an adjacency matrix A for a chain graph with $(A_{ij} = A_{ji} = 1)$ representing an undirected edge between V_i and V_j and $(A_{ij} = 1, A_{ji} = 0)$ representing a directed edge from V_i to V_j. Moreover, it is not difficult to see that $\mathbb{E}[\text{vertex degree}] = N$, where an adjacent vertex can be linked by either an undirected or a directed edge.

Given a randomly generated chain graph G with ordered chain components $C_1, \ldots, C_k$, we generate a Gaussian distribution on it via the rnorm.cg function from the LCD R package.

Metrics for Evaluation We evaluate the performance of the proposed algorithms in terms of the six measurements that are commonly used (Colombo and Maathuis, 2014; Ma, Xie, and Geng, 2008; Tsamardinos et al., 2006) for constraint-based learning algorithms: (a) the true positive rate (TPR) (also known as sensitivity, recall, and hit rate), (b) the false positive rate (FPR) (also known as fall-out), (c) the true discovery rate (TDR) (also known as precision or positive predictive value), (d) accuracy (ACC) for the skeleton, (e) the structural Hamming distance (SHD) (this is the metric described in Tsamardinos et al. (2006) to compare the structure of the learned and the original graphs), and (f) run-time for the pattern recovery algorithms. In short, $TPR = \frac{\text{true positive } (TP)}{\text{the number of real positive cases in the data } (Pos)}$ is the ratio of the number of correctly identified edges over total number of edges, $FPR = \frac{\text{false positive } (FP)}{\text{the number of real negative cases in the data } (Neg)}$ is the ratio of the number of incorrectly identified edges over total number of gaps, $TDR = \frac{\text{true positive } (TP)}{\text{the total number of edges in the recovered CG}}$ is the ratio of the number of correctly identified edges over total number of edges (both in estimated graph), $ACC = \frac{\text{true positive } (TP) + \text{ true negative } (TN)}{Pos+Neg}$, and SHD is the number of legitimate operations needed to change the current resulting graph to the true CG, where legitimate operations are: (a) add or delete an edge and (b) insert, delete or reverse an edge orientation. In principle, a large TPR, TDR, and ACC, a small FPR and SHD indicate good performance.

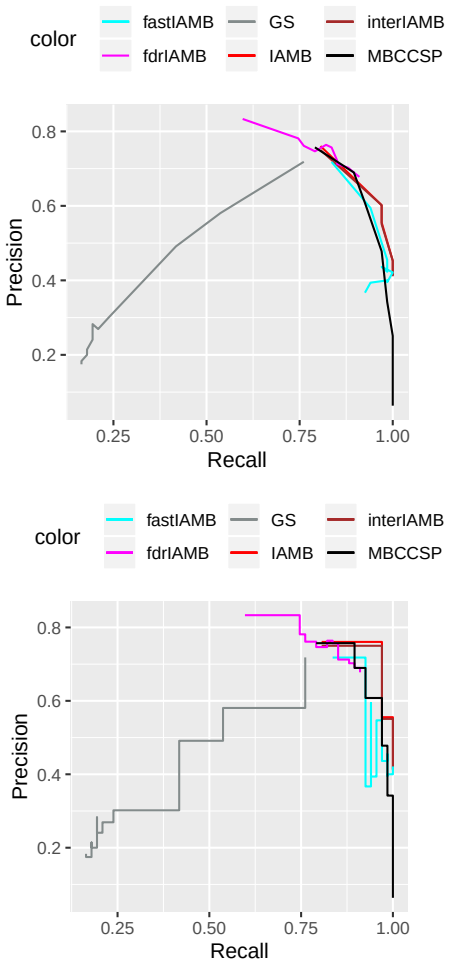

Figure 6: Precision-Recall ROC curves (pathwise and stepwise) along several different alpha values ($\alpha \in (0.005, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$) for a randomly generated Gaussian CG model with 50 variables and N = 3; these curves show the precision-recall trade-off.

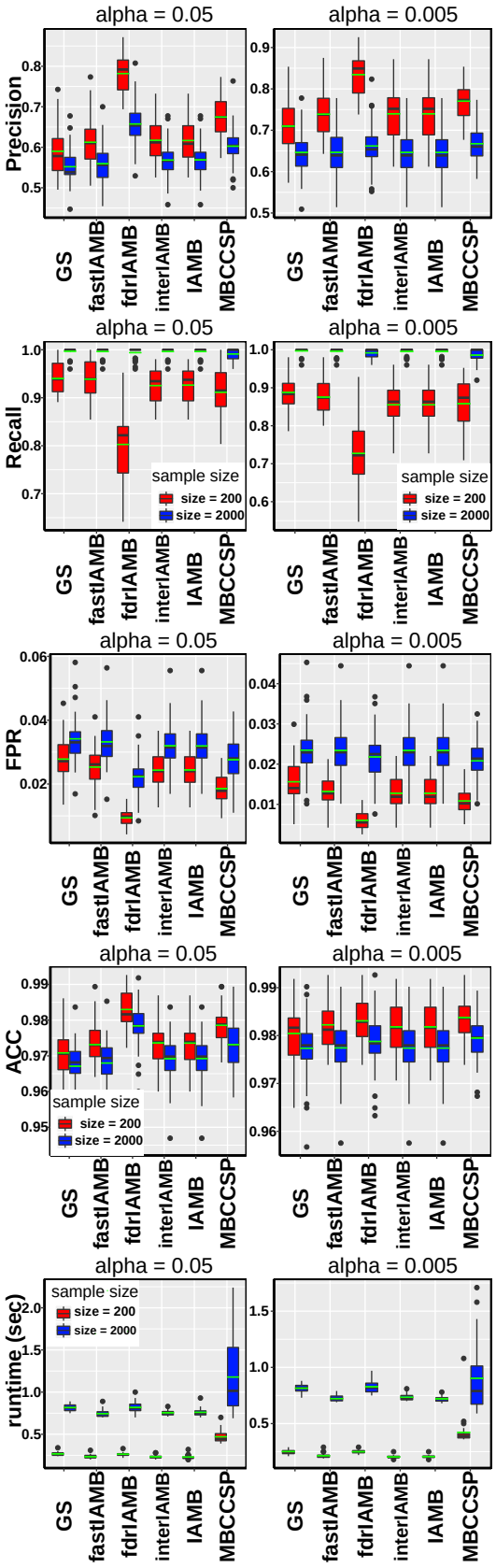

Figure 7: Performance of Markov blanket recovery algorithms for randomly generated Gaussian chain graph models: over 30 repetitions with 50 variables correspond to N = 2. The green line in a box indicates the mean of that group.

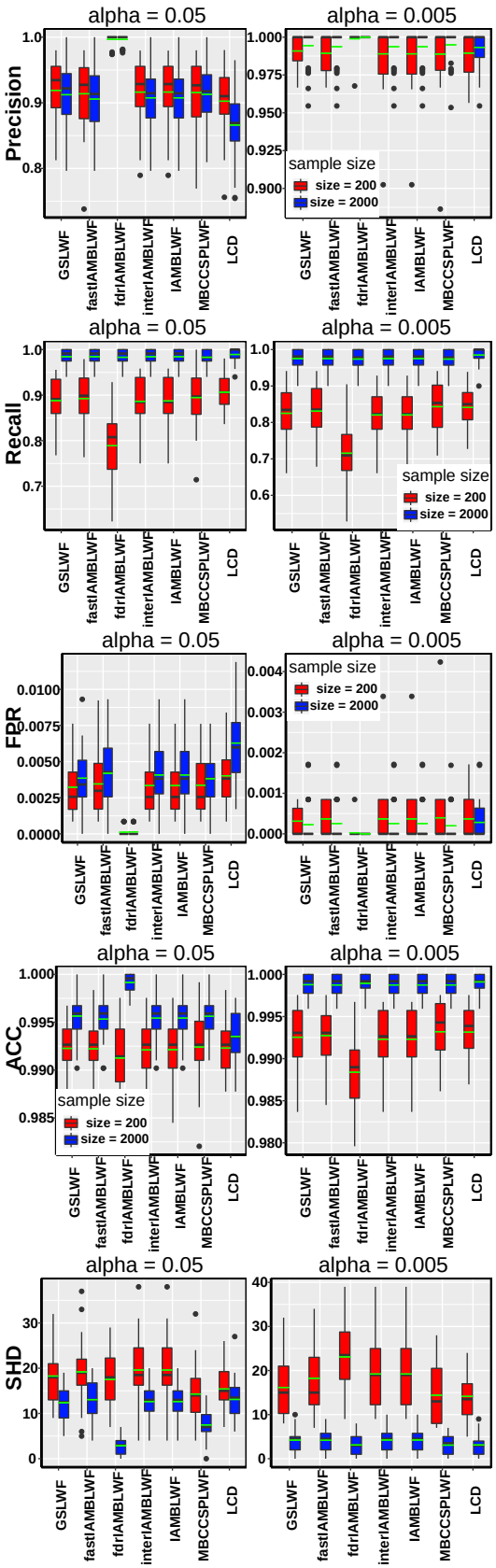

Figure 8: Performance of LCD and MbLWF algorithms for randomly generated Gaussian chain graph models: over 30 repetitions with 50 variables correspond to N = 2. The green line in a box indicates the mean of that group.

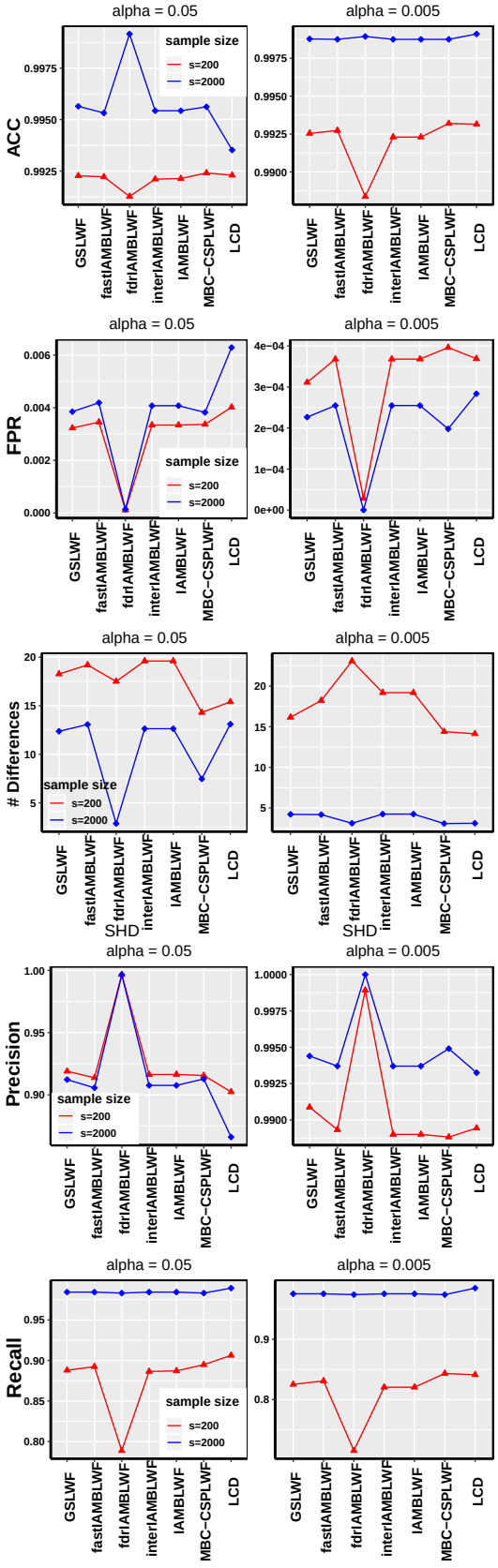

Figure 9: Performance of LCD and MbLWF algorithms for randomly generated Gaussian chain graph models: average over 30 repetitions with 50 variables correspond to N = 2.

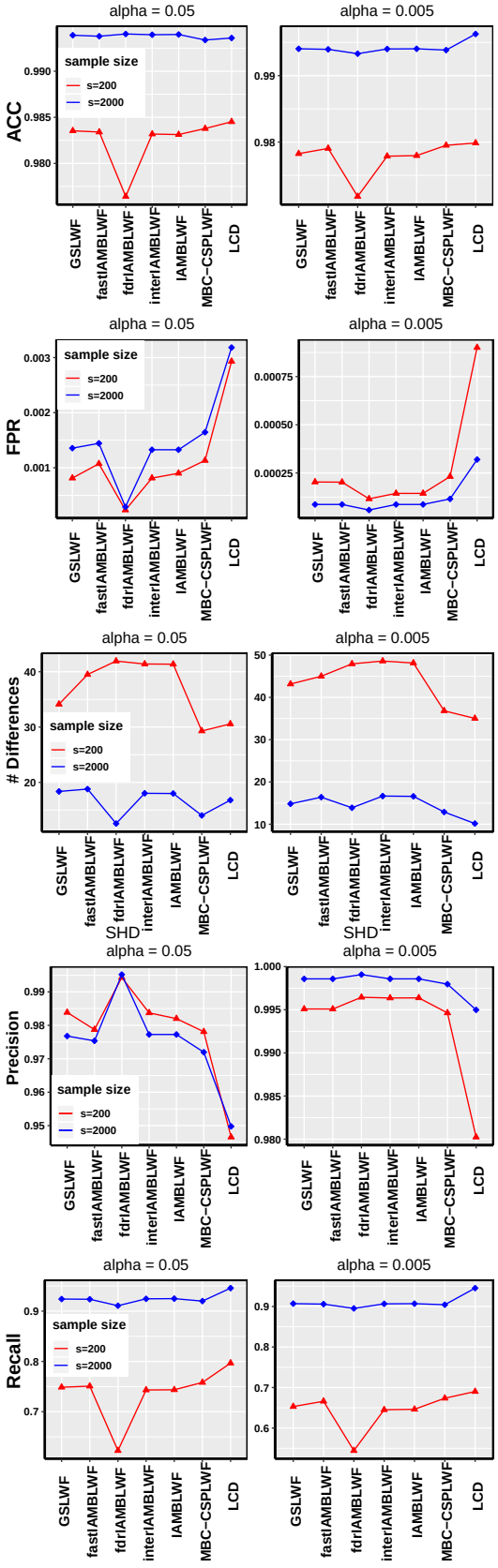

Figure 10: Performance of LCD and MbLWF algorithms for randomly generated Gaussian chain graph models: average over 30 repetitions with 50 variables correspond to N = 3.