LEARNING MARGINAL AMP CHAIN GRAPHS UNDER FAITHFULNESS REVISITED

JOSE M. PEÑA
ADIT, IDA
LINKÖPING UNIVERSITY, SWEDEN
JOSE.M.PENA@LIU.SE

MANUEL GÓMEZ-OLMEDO
DEPT. COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE
UNIVERSITY OF GRANADA, SPAIN
MGOMEZ@DECSAI.UGR.ES

Abstract. Marginal AMP chain graphs are a recently introduced family of models that is based on graphs that may have undirected, directed and bidirected edges. They unify and generalize the AMP and the multivariate regression interpretations of chain graphs. In this paper, we present a constraint based algorithm for learning a marginal AMP chain graph from a probability distribution which is faithful to it. We show that the marginal AMP chain graph returned by our algorithm is a distinguished member of its Markov equivalence class.

We show that our algorithm performs well in practice. Specifically, we sample probability distributions that are faithful to directed and acyclic graphs, which are a subfamily of marginal AMP chain graphs, and show that our algorithm performs relatively well compared to an algorithm tailored to learning directed and acyclic graphs.

Finally, we show that the extension of Meek's conjecture to marginal AMP chain graphs does not hold, which compromises the development of efficient and correct score+search learning algorithms under assumptions weaker than faithfulness.

1. Introduction

Chain graphs (CGs) are graphs with possibly directed and undirected edges, and no semidirected cycle. They have been extensively studied as a formalism to represent independence models, because they can model symmetric and asymmetric relationships between the random variables of interest. However, there are three different interpretations of CGs as independence models: The Lauritzen-Wermuth-Frydenberg (LWF) interpretation (Lauritzen, 1996), the multivariate regression (MVR) interpretation (Cox and Wermuth, 1996), and the Andersson-Madigan-Perlman (AMP) interpretation (Andersson et al., 2001). It is worth mentioning that no interpretation subsumes another: There are many independence models that can be represented by a CG under one interpretation but that cannot be represented by any CG under the other interpretations (Andersson et al., 2001; Sonntag and Peña, 2015). Moreover, although MVR CGs were originally represented using dashed directed and undirected edges, we like other authors prefer to represent them using solid directed and bidirected edges.

Recently, a new family of models has been proposed to unify and generalize the AMP and MVR interpretations of CGs (Peña, 2014b). This new family, named marginal AMP (MAMP) CGs, is based on graphs that may have undirected, directed and bidirected edges. This paper complements that by Peña (2014b) by presenting an algorithm for learning a MAMP CG from a probability distribution which is faithful to it. Our algorithm is constraint based and builds upon those developed by Sonntag and Peña (2012) and Peña (2014a) for learning, respectively, MVR and AMP CGs under the faithfulness assumption. It is worth mentioning that there also exist algorithms for learning LWF CGs under the faithfulness assumption (Ma

Date: pgm2014jmpfollowup4.tex, 09:44, 19/12/14.

et al., 2008; Studený, 1997) and under the milder composition property assumption (Peña et al., 2014). In this paper, we also show that the extension of Meek's conjecture to MAMP CGs does not hold, which compromises the development of efficient and correct score+search learning algorithms under assumptions weaker than faithfulness.

Finally, we should mention that this paper is an extended version of that by Peña (2014c). The extension consists in that the learning algorithm presented in that paper has been modified so that it returns a distinguished member of a Markov equivalence class of MAMP CGs, rather than just a member of the class. As a consequence, the proof of correctness of the algorithm has changed significantly. Moreover, the algorithm has been implemented and evaluated. This paper reports the results of the evaluation for the first time. Specifically, we sampled probability distributions that were faithful to DAGs, which are a subfamily of MAMP CGs, and confirmed that our algorithm performs relatively well compared to the algorithm proposed by Meek (1995) which, unlike ours, is tailored to learning DAGs.

The rest of this paper is organized as follows. We start with some preliminaries in Section 2. Then, we introduce MAMP CGs in Section 3, followed by the algorithm for learning them in Section 4. In that section, we also include a review of other learning algorithms that are related to ours. We report the experimental results in Section 5. We close the paper with some discussion in Section 6. All the proofs appear in an appendix at the end of the paper.

2. Preliminaries

In this section, we introduce some concepts of models based on graphs, i.e. graphical models. Most of these concepts have a unique definition in the literature. However, a few concepts have more than one and we opt for the most suitable in this work. All the graphs and probability distributions in this paper are defined over a finite set V. All the graphs in this paper are simple, i.e. they contain at most one edge between any pair of nodes. The elements of V are not distinguished from singletons.

If a graph G contains an undirected, directed or bidirected edge between two nodes V_1 and V_2, then we write that $V_1 - V_2$, $V_1 \rightarrow V_2$ or $V_1 \leftrightarrow V_2$ is in G. We represent with a circle, such as in $V_1 \circ\!\!\rightarrow V_2$ or $V_1 \circ\!\!-\!\!\circ V_2$, that the end of an edge is unspecified, i.e. it may be an arrowhead or nothing. If the edge is of the form $V_1 \circ\!\!\rightarrow V_2$, then we say it has an arrowhead at V_2. If the edge is of the form $V_1 \rightarrow V_2$, then we say that it has an arrowtail at V_1. The parents of a set of nodes X of G is the set $pa_G(X) = \{V_1 | V_1 \rightarrow V_2 \text{ is in } G, V_1 \notin X \text{ and } V_2 \in X\}$. The children of X is the set $ch_G(X) = \{V_1 | V_1 \leftarrow V_2 \text{ is in } G, V_1 \notin X \text{ and } V_2 \in X\}$. The neighbors of X is the set $ne_G(X) = \{V_1 | V_1 - V_2 \text{ is in } G, V_1 \notin X \text{ and } V_2 \in X\}$. The spouses of X is the set $sp_G(X) = \{V_1 | V_1 \leftrightarrow V_2 \text{ is in } G, V_1 \notin X \text{ and } V_2 \in X\}$. The adjacents of X is the set $ad_G(X) = ne_G(X) \cup pa_G(X) \cup ch_G(X) \cup sp_G(X)$. A route between a node V_1 and a node V_n in G is a sequence of (not necessarily distinct) nodes $V_1, \dots, V_n$ such that $V_i \in ad_G(V_{i+1})$ for all $1 \leq i < n$. If the nodes in the route are all distinct, then the route is called a path. The length of a route is the number of (not necessarily distinct) edges in the route, e.g. the length of the route $V_1, \dots, V_n$ is $n-1$. A route is called descending if $V_i \rightarrow V_{i+1}$, $V_i - V_{i+1}$ or $V_i \leftrightarrow V_{i+1}$ is in G for all $1 \leq i < n$. A route is called strictly descending if $V_i \rightarrow V_{i+1}$ is in G for all $1 \leq i < n$. The descendants of a set of nodes X of G is the set $de_G(X) = \{V_n |$ there is a descending route from V_1 to V_n in G, $V_1 \in X$ and $V_n \notin X\}$. The strict ascendants of X is the set $san_G(X) = \{V_1 |$ there is a strictly descending route from V_1 to V_n in G, $V_1 \notin X$ and $V_n \in X\}$. A route $V_1, \dots, V_n$ in G is called a cycle if $V_n = V_1$. Moreover, it is called a semidirected cycle if $V_n = V_1$, $V_1 \rightarrow V_2$ is in G and $V_i \rightarrow V_{i+1}$, $V_i \leftrightarrow V_{i+1}$ or $V_i - V_{i+1}$ is in G for all $1 < i < n$. A cycle has a chord if two non-consecutive nodes of the cycle are adjacent in G. The subgraph of G induced by a set of nodes X is the graph over X that has all and only the edges in G whose both ends are in X. Moreover, a triplex $(\{A, C\}, B)$ in G is an induced subgraph of the form $A \circ\!\!\rightarrow B \leftarrow\!\!\circ C$, $A \circ\!\!\rightarrow B - C$ or $A - B \leftarrow\!\!\circ C$.

A directed and acyclic graph (DAG) is a graph with only directed edges and without semidirected cycles. An AMP chain graph (AMP CG) is a graph whose every edge is directed or undirected such that it has no semidirected cycles. A MVR chain graph (MVR CG) is a graph whose every edge is directed or bidirected such that it has no semidirected cycles. Clearly, DAGs are a special case of AMP and MVR CGs: DAGs are AMP CGs without undirected edges, and DAGs are MVR CGs without bidirected edges. We now recall the semantics of AMP and MVR CGs. A node B in a path ρ in an AMP CG G is called a triplex node in ρ if $A \rightarrow B \leftarrow C$, $A \rightarrow B - C$, or $A - B \leftarrow C$ is a subpath of ρ. Moreover, ρ is said to be Z-open with $Z \subseteq V$ when

- every triplex node in ρ is in $Z \cup san_G(Z)$, and
- every non-triplex node B in ρ is outside Z, unless $A - B - C$ is a subpath of ρ and $pa_G(B) \setminus Z \neq \emptyset$.

A node B in a path ρ in an MVR CG G is called a triplex node in ρ if $A \multimap\!\!\rightarrow B \leftarrow\!\!\multimap C$ is a subpath of ρ. Moreover, ρ is said to be Z-open with $Z \subseteq V$ when

- every triplex node in ρ is in $Z \cup san_G(Z)$, and
- every non-triplex node B in ρ is outside Z.

Let X, Y and Z denote three disjoint subsets of V. When there is no Z-open path in an AMP or MVR CG G between a node in X and a node in Y, we say that X is separated from Y given Z in G and denote it as $X \perp_G Y|Z$. The independence model represented by G, denoted as $I(G)$, is the set of separations $X \perp_G Y|Z$. In general, $I(G)$ is different depending on whether G is an AMP or MVR CG. However, it is the same when G is a DAG.

3. MAMP CGs

In this section, we review marginal AMP (MAMP) CGs. We refer the reader to the work by Peña (2014b) for more details. Specifically, a graph G containing possibly directed, bidirected and undirected edges is a MAMP CG if

C1. G has no semidirected cycle,

C2. G has no cycle $V_1, \ldots, V_n = V_1$ such that $V_1 \leftrightarrow V_2$ is in G and $V_i - V_{i+1}$ is in G for all $1 < i < n$, and

C3. if $V_1 - V_2 - V_3$ is in G and $sp_G(V_2) \neq \emptyset$, then $V_1 - V_3$ is in G too.

The semantics of MAMP CGs is as follows. A node B in a path ρ in a MAMP CG G is called a triplex node in ρ if $A \multimap\!\!\rightarrow B \leftarrow\!\!\multimap C$, $A \multimap\!\!\rightarrow B - C$, or $A - B \leftarrow\!\!\multimap C$ is a subpath of ρ. Moreover, ρ is said to be Z-open with $Z \subseteq V$ when

- every triplex node in ρ is in $Z \cup san_G(Z)$, and
- every non-triplex node B in ρ is outside Z, unless $A - B - C$ is a subpath of ρ and $sp_G(B) \neq \emptyset$ or $pa_G(B) \setminus Z \neq \emptyset$.

Let X, Y and Z denote three disjoint subsets of V. When there is no Z-open path in G between a node in X and a node in Y, we say that X is separated from Y given Z in G and denote it as $X \perp_G Y|Z$. The independence model represented by G, denoted as $I(G)$, is the set of separations $X \perp_G Y|Z$. We denote by $X \perp_p Y|Z$ (respectively $X \not\perp_p Y|Z$) that X is independent (respectively dependent) of Y given Z in a probability distribution p. We say that p is faithful to G when $X \perp_p Y|Z$ if and only if $X \perp_G Y|Z$ for all X, Y and Z disjoint subsets of V. We say that two MAMP CGs are Markov equivalent if they represent the same independence model. We also say that two MAMP CGs are triplex equivalent if they have the same adjacencies and the same triplexes. Two MAMP CGs are Markov equivalent if and only if they are triplex equivalent (Peña, 2014b, Theorem 7).

Clearly, AMP and MVR CGs are special cases of MAMP CGs: AMP CGs are MAMP CGs without bidirected edges, and MVR CGs are MAMP CGs without undirected edges. Then, the union of AMP and MVR CGs is a subfamily of MAMP CGs. The following example shows that it is actually a proper subfamily.

Example 1. *The independence model represented by the MAMP CG G below cannot be represented by any AMP or MVR CG.*

$$\begin{tikzcd} A \arrow[r] & B \arrow[r, no head] \arrow[d, no head] & C \arrow[d, leftrightarrow] \\ & D \arrow[r, leftrightarrow] & E \end{tikzcd}$$

To see it, assume to the contrary that it can be represented by an AMP CG H. Note that H is a MAMP CG too. Then, G and H must have the same triplexes. Then, H must have triplexes $(\{A, D\}, B)$ and $(\{A, C\}, B)$ but no triplex $(\{C, D\}, B)$. So, $C - B - D$ must be in H. Moreover, H must have a triplex $(\{B, E\}, C)$. So, $C \leftarrow E$ must be in H. However, this implies that H does not have a triplex $(\{C, D\}, E)$, which is a contradiction because G has such a triplex. To see that no MVR CG can represent the independence model represented by G, simply note that no MVR CG can have triplexes $(\{A, D\}, B)$ and $(\{A, C\}, B)$ but no triplex $(\{C, D\}, B)$.

Finally, it is worth mentioning that MAMP CGs are not the first family of models to be based on graphs that may contain undirected, directed and bidirected edges. Other such families are summary graphs after replacing the dashed undirected edges with bidirected edges (Cox and Wermuth, 1996), MC graphs (Koster, 2002), maximal ancestral graphs (Richardson and Spirtes, 2002), and loopless mixed graphs (Sadeghi and Lauritzen, 2014). However, the separation criteria for these families are identical to that of MVR CGs. Then, MVR CGs are a subfamily of these families but AMP CGs are not. For further details, see also the works by Richardson and Spirtes (2002, p. 1025) and Sadeghi and Lauritzen (2014, Sections 4.1-4.3). Therefore, MAMP CGs are the only graphical models in the literature that generalize both AMP and MVR CGs.

4. Algorithm for Learning MAMP CGs

In this section, we present our algorithm for learning a MAMP CG from a probability distribution which is faithful to it. Prior to that, we describe how we represent a class of Markov equivalent MAMP CGs, because that is the output of the algorithm. In the works by Andersson et al. (1997), Andersson and Perlman (2006), Meek (1995), Sonntag and Peña (2015) and Sonntag et al. (2015), the authors define the unique representant of a class of Markov equivalent DAGs, AMP CGs and MVR CGs as the graph H such that (i) H has the same adjacencies as every member of the class, and (ii) H has an arrowhead at an edge end if and only if there is a member of the class with an arrowhead at that edge end and there is no member of the class with a arrowtail at that edge end. Clearly, this definition can also be used to construct a unique representant of a class of Markov equivalent MAMP CGs. We call the unique representant of a class of Markov equivalent DAGs, AMP CGs, MVR CGs or MAMP CGs the essential graph (EG) of the class. We show below that the EG of a class of Markov equivalent MAMP CGs is always a member of the class. The EG of a class of Markov equivalent AMP CGs also has this desirable feature (Andersson and Perlman, 2006, Theorem 3.2), but the EG of a class of Markov equivalent DAGs or MVR CGs does not (Andersson et al., 1997; Sonntag et al., 2015).

Now, we present our algorithm for learning MAMP CGs under the faithfulness assumption. The algorithm can be seen in Table 1. Note that the algorithm returns the EG of a class of Markov equivalent MAMP CGs. The correctness of the algorithm is proven in the appendix. Our algorithm builds upon those developed by Meek (1995), Peña (2014a), Sonntag and Peña (2012) and Spirtes et al. (1993) for learning DAGs, AMP CGs and MVR CGs under the faithfulness assumption. Like theirs, our algorithm consists of two phases: The first

Table 1. Algorithm for learning MAMP CGs.

Input: A probability distribution p that is faithful to an unknown MAMP CG G.
Output: The EG H of the Markov equivalence class of G.

1 Let H denote the complete undirected graph
2 Set $l = 0$
3 Repeat while $l \leq |V| - 2$
4   For each ordered pair of nodes A and B in H st $A \in ad_H(B)$ and $|[ad_H(A) \cup ad_H(ad_H(A))] \setminus \{A, B\}| \geq l$
5   If there is some $S \subseteq [ad_H(A) \cup ad_H(ad_H(A))] \setminus \{A, B\}$ st $|S| = l$ and $A \perp_p B | S$ then
6   Set $S_{AB} = S_{BA} = S$
7   Remove the edge $A - B$ from H
8   Set $l = l + 1$
9 Apply the rules R1-R4 to H while possible
10 Replace every edge $A - B$ in every cycle in H that is of length greater than three, chordless, and without blocks with $A \vdash\!\dashv B$
11 Apply the rules R2-R4 to H while possible
12 Replace every edge $A \vdash B$ in H with $A \to B$
13 Replace every edge $A \vdash\!\dashv B$ in H with $A \leftrightarrow B$
14 Replace every induced subgraph $A \leftrightarrow B \leftrightarrow C$ in H st $B \in S_{AC}$ with $A - B - C$
15 If H has an induced subgraph $\begin{tikzcd} A \arrow[r, leftrightarrow] \arrow[rr, bend left, dash] & B \arrow[r, dash] & C \end{tikzcd}$ then
16   Replace the edge $A \leftrightarrow B$ in H with $A - B$
17   Go to line 15
18 Return H

Table 2. Rules R1-R4.

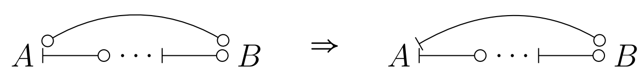
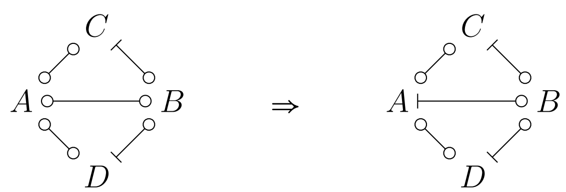
R1	$A \circ\!\!-\!\!\circ B \circ\!\!-\!\!\circ C$ $\wedge\ B \notin S_{AC}$	$\Rightarrow$	$A \vdash\!\!-\!\!\circ B \circ\!\!-\!\!\dashv C$
R2	$A \vdash\!\!-\!\!\circ B \circ\!\!-\!\!\circ C$ $\wedge\ B \in S_{AC}$	$\Rightarrow$	$A \vdash\!\!-\!\!\circ B \vdash\!\!-\!\!\circ C$
R3			
R4	 $\wedge\ A \in S_{CD}$		

Table 3. Algorithm for learning AMP CGs presented by Peña (2014a).

Input: A probability distribution p that is faithful to an unknown AMP CG G.
Output: The EG CG H of the Markov equivalence class of G.

1 Let H denote the complete undirected graph
2 Set $l = 0$
3 Repeat while $l \leq |V| - 2$
4   For each ordered pair of nodes A and B in H st $A \in ad_H(B)$ and $|[ad_H(A) \cup ad_H(ad_H(A))] \setminus \{A, B\}| \geq l$
5   If there is some $S \subseteq [ad_H(A) \cup ad_H(ad_H(A))] \setminus \{A, B\}$ st $|S| = l$ and $A \perp_p B | S$ then
6   Set $S_{AB} = S_{BA} = S$
7   Remove the edge $A - B$ from H
8   Set $l = l + 1$
9 Apply the rules R1-R4 to H while possible
10 Replace every edge $A - B$ in every cycle in H that is of length greater than three, chordless, and without blocks with $A \vdash\!\dashv B$
11 Apply the rules R2-R4 to H while possible
12 Replace every edge $A \vdash B$ in H with $A \to B$
13 Replace every edge $A \vdash\!\dashv B$ in H with $A - B$
14 Return H

phase (lines 1-8) aims at learning adjacencies, whereas the second phase (lines 9-17) aims at directing some of the adjacencies learnt. Specifically, the first phase declares that two nodes are adjacent if and only if they are not separated by any set of nodes. Note that the algorithm does not test every possible separator (see line 5). Note also that the separators tested are tested in increasing order of size (see lines 2, 5 and 8). The second phase consists of two steps. In the first step (lines 9-11), the ends of some of the edges learnt in the first phase are blocked according to the rules R1-R4 in Table 2. A block is represented by a perpendicular line at the edge end such as in $\vdash$ or $\vdash\!\dashv$, and it means that the edge cannot be a directed edge pointing in the direction of the block. Note that $\vdash\!\dashv$ does not mean that the edge must be undirected: It means that the edge cannot be a directed edge in either direction and, thus, it must be a bidirected or undirected edge. In the second step (lines 12-17), some edges get directed. Specifically, the edges with exactly one unblocked end get directed in the direction of the unblocked end (see line 12). The rest of the edges get bidirected (see line 13), unless this produces a false triplex (see line 14) or violates the constraint C2 (see lines 15-17). Note that only cycles of length three are checked for the violation of the constraint C2.

The rules R1-R4 in Table 2 work as follows: If the conditions in the antecedent of a rule are satisfied, then the modifications in the consequent of the rule are applied. Note that the ends of some of the edges in the rules are labeled with a circle such as in $\vdash\!\circ$ or $\circ\!-\!\circ$. The circle represents an unspecified end, i.e. a block or nothing. The modifications in the consequents of the rules consist in adding some blocks. Note that only the blocks that appear in the consequents are added, i.e. the circled ends do not get modified. The conditions in the antecedents of R1, R2 and R4 consist of an induced subgraph of H and the fact that some of its nodes are or are not in some separators found in line 6. The condition in the antecedent of R3 consists of just an induced subgraph of H. Specifically, the antecedent says that there is a cycle in H whose edges have certain blocks. Note that the cycle must be chordless.

4.1. **Related Algorithms.** In this section, we review two algorithms for learning AMP CGs and DAGs that can be seen as particular cases of the algorithm presented above.

Table 4. Rules R1'-R4'.

R1'	$A - B - C$ $\wedge\ B \notin S_{AC}$	$\Rightarrow$	$A \to B \leftarrow C$
R2'	$A \to B - C$ $\wedge\ B \in S_{AC}$	$\Rightarrow$	$A \to B \to C$
R3'	$\begin{tikzcd} A \arrow[r] \arrow[rr, bend left, dash] & \cdots \arrow[r] & B \end{tikzcd}$	$\Rightarrow$	$\begin{tikzcd} A \arrow[r] \arrow[rr, bend left] & \cdots \arrow[r] & B \end{tikzcd}$
R4'	$\begin{tikzcd} & C \arrow[dr] & \\ A \arrow[ur, Circle-Circle] \arrow[rr, dash] \arrow[dr, Circle-Circle] & & B \\ & D \arrow[ur] & \end{tikzcd}$ $\wedge\ A \in S_{CD}$	$\Rightarrow$	$\begin{tikzcd} & C \arrow[dr] & \\ A \arrow[ur, Circle-Circle] \arrow[rr] \arrow[dr, Circle-Circle] & & B \\ & D \arrow[ur] & \end{tikzcd}$

Table 5. Rules R3" and R4".

R3"	$\begin{tikzcd} A \arrow[r] \arrow[rr, bend left, dash] & B \arrow[r] & C \end{tikzcd}$	$\Rightarrow$	$\begin{tikzcd} A \arrow[r] \arrow[rr, bend left] & B \arrow[r] & C \end{tikzcd}$
R4"	$\begin{tikzcd} & C \arrow[dr] & \\ A \arrow[ur, dash] \arrow[rr, dash] \arrow[dr, dash] & & B \\ & D \arrow[ur] & \end{tikzcd}$ $\wedge\ A \in S_{CD}$	$\Rightarrow$	$\begin{tikzcd} & C \arrow[dr] & \\ A \arrow[ur, dash] \arrow[rr] \arrow[dr, dash] & & B \\ & D \arrow[ur] & \end{tikzcd}$

4.1.1. *Algorithm for Learning AMP CGs.* Peña (2014a) presents an algorithm for learning AMP CGs under the faithfulness assumption. We show next that that algorithm coincides with the algorithm for learning MAMP CGs in Table 1 when G is an AMP CG. Specifically, if G is an AMP CG then it only has directed and undirected edges and, thus, any edge $A \mapsto B$ in H corresponds to an edge $A - B$ in G. Therefore, line 13 in Table 1 should be modified accordingly. After this modification, lines 14-17 do not make sense and, thus, they can be removed. The resulting algorithm can be seen in Table 3. This is exactly the algorithm for learning AMP CGs presented by Peña (2014a), except for lines 10-11. Adding these lines ensures that the output is the EG of the Markov equivalence class of G and not just a CG in class (Sonntag and Peña, 2015, Theorem 11).

Table 6. Algorithm for learning DAGs presented by Meek (1995).

Input: A probability distribution p that is faithful to an unknown DAG G.
Output: The EG CG H of the Markov equivalence class of G.

1 Let H denote the complete undirected graph
2 Set $l = 0$
3 Repeat while $l \leq |V| - 2$
4   For each ordered pair of nodes A and B in H st $A \in ad_H(B)$ and $|ad_H(A) \setminus \{B\}| \geq l$
5   If there is some $S \subseteq ad_H(A) \setminus \{B\}$ st $|S| = l$ and $A \perp_p B | S$ then
6   Set $S_{AB} = S_{BA} = S$
7   Remove the edge $A - B$ from H
8   Set $l = l + 1$
9 Apply the rules R1', R2', R3" and R4" to H while possible
10 Return H

4.1.2. *Algorithm for Learning DAGs.* Meek (1995) presents an algorithm for learning DAGs under the faithfulness assumption. We show next that that algorithm coincides with the algorithm for learning AMP CGs in Table 3 when G is a DAG.

Firstly, the set of nodes $[ad_H(A) \cup ad_H(ad_H(A))] \setminus \{A, B\}$ is considered in lines 4 and 5 so as to guarantee that G and H have the same adjacencies after line 8, as proven in Lemma 1 in the appendix. However, if G is a DAG then it only has directed edges and, thus, the proof of the lemma simplifies so that it suffices to consider $ad_H(A) \setminus \{B\}$ in lines 4 and 5 to guarantee that G and H have the same adjacencies after line 8. Thus, if G is a DAG then we can replace $[ad_H(A) \cup ad_H(ad_H(A))] \setminus \{A, B\}$ in lines 4 and 5 with $ad_H(A) \setminus \{B\}$.

Secondly, if G is a DAG then it only has directed edges and, thus, H cannot have any edge $A \mapsto\!\!\!\!\leftarrow B$. Therefore, lines 10, 11 and 13 in Table 3 do not make sense and, thus, they can be removed. Moreover, if H cannot have any edge $A \mapsto\!\!\!\!\leftarrow B$, then any edge $A \vdash\!\!\circ B$ in the rules R1-R4 in Table 2 can be replaced by $A \vdash B$. For the same reason, any modification of an edge $A \circ\!\!-\!\!\circ B$ into $A \vdash\!\!\circ B$ in the rules can be replaced by a modification of an edge $A - B$ into $A \vdash B$. These observations together with line 12 in Table 3 imply that the rules R1-R4 can be rewritten as the rules R1'-R4' in Table 4. After this rewriting, line 12 in Table 3 is not needed anymore and, thus, it can be removed.

Finally, if G is a DAG, then the rule R3' in Table 4 does not need to be applied to cycles of length greater than three. To see it, assume that the rule is applied to the cycle $A \to V_1 \to \cdots \to V_n \to B - A$ in H with $n > 1$. Recall that the cycle must be chordless. Then, the rule modifies the edge $A - B$ in H into $A \to B$. This implies that G has an induced subgraph $A \to B \leftarrow V_n$, i.e. G has a triplex $(\{A, V_n\}, B)$. Clearly, $B \notin S_{AV_n}$. Then, the edge $A - B$ in H would have been modified into $A \to B$ by the rule R1' anyway. Therefore, the rule R3' does not need to be applied to cycles of length greater than three and, thus, it can be replaced by the rule R3" in Table 5. Likewise, the rule R4' in Table 4 can be replaced by the rule R4" in Table 5. To see it, note that $A \in S_{CD}$ implies that G has an induced subgraph $C \to A \to D$, $C \leftarrow A \leftarrow D$ or $C \leftarrow A \to D$. Therefore, if R4' can be applied but R4" cannot, then H must have an induced subgraph $A \to C \to B - A$ or $A \to D \to B - A$. Then, the edge $A - B$ gets modified into $A \to B$ by R3". The resulting algorithm can be seen in Table 6. This is exactly the algorithm for learning DAGs presented by Meek (1995), except for the names of the rules.

5. Experiments

In this section, we compare the performance of our algorithm (Table 1) and that by Meek (1995) (Table 6) on samples from probability distributions that are faithful to DAGs. Therefore, the learning data is tailored to Meek's algorithm rather than to ours. Of course, both algorithms should perform similarly on the large sample limit, because DAGs are a subfamily of MAMP CGs. We are however interested in their relative performance for medium size samples. Since our algorithm tests in line 5 all the separators that Meek's algorithm tests and many more, we expect that our algorithm drops more edges in line 7 and, thus, the edges retained are more likely to be true positive. Therefore, we expect that our algorithm shows lower recall but higher precision between the adjacencies in the sampled and learnt graphs. We also expect that our algorithm shows lower recall between the triplexes in the sampled and learnt graphs because, as discussed before, the edges involved in a true triplex are more likely to be dropped in our algorithm. Likewise, we expect that our algorithm shows lower precision between the triplexes in the sampled and learnt graphs, because dropping a true edge may create a false positive triplex, and this is more likely to happen in our algorithm. In this section, we try to elucidate the extend of this decrease in the performance of our algorithm compared to Meek's. Another way to look at this question is by noting that, despite the sampled graph belongs to the search spaces considered by both algorithms, our algorithm considers a much bigger search space, which implies a larger risk of ending in a suboptimal solution. For instance, the ratio of the numbers of independence models representable by an AMP or MVR CG and those representable by a DAG is approximately 7 for 8 nodes, 26 for 11 nodes, and 1672 for 20 nodes (Sonntag and Peña, 2015). Note that the ratio of the numbers of independence models representable by a MAMP CG and those representable by a DAG is much larger than the figures given (recall Example 1). In this section, we try to elucidate the effects that this larger search space has on the performance of our algorithm.

We implemented the algorithms in R. We implemented the algorithms as they appear in Tables 1 and Table 6 and, thus, we did not implement any conflict resolution technique such as those discussed by Ramsey et al. (2006), Cano et al. (2008), and Colombo and Maathuis (2014). The implementation will be made publicly available. To obtain sample versions of the algorithms, we replaced $A \perp_p B | S$ in line 5 with a hypothesis test. Specifically, we used the default test implemented by the function `ci.test` of the R package `bnlearn`. This is a test based on mutual information, whose statistic asymptotically follows a χ^2 distribution with the appropriate degrees of freedom. We used the default significance level 0.05.

In the experiments, we considered the following Bayesian networks: Asia (8 nodes and 8 edges), Sachs (11 nodes and 17 edges), Child (20 nodes and 17 edges), Insurance (27 nodes and 56 edges), Mildew (35 nodes and 46 edges), Alarm (37 nodes and 46 edges) and Barley (48 nodes and 84 edges). All the networks were obtained from the repository at `www.bnlearn.com`. All the nodes in these networks represent discrete random variables. From each network considered, we obtained 10 samples of size 500, 1000, 5000, 10000 and 50000. For each sampled size, we ran our and Meek's algorithms on the corresponding 10 samples and, then, computed the average precision and recall between the adjacencies in the sampled and learnt graphs, as well as between the triplexes in the sampled and learnt graphs. We denote the former two measures as PA and RA, and the latter two as PT and RT. We chose these measures because it is the adjacencies and triplexes what determines the Markov equivalence class of the DAG or MAMP CG learnt (recall Section 3).

Tables 7-13 show the results of our experiments. Broadly speaking, the results confirm our expectation of our algorithm scoring lower RA, RT and PT, and higher PA. However, there are a few cases where the results depart from the expectation and that indicate that our algorithm performs slightly better than expected:

- For the Asia network, our algorithm scores better RT and PT for all the sample sizes.

Table 7. Results for the Asia network (8 nodes and 8 edges).

	size	500	1000	5000	10000	50000
Our algorithm	RA	0.50 ± 0.00	0.53 ± 0.05	0.51 ± 0.04	0.53 ± 0.05	0.62 ± 0.00
	PA	0.98 ± 0.06	0.96 ± 0.08	1.00 ± 0.00	0.98 ± 0.06	1.00 ± 0.00
	RT	0.08 ± 0.18	0.22 ± 0.24	0.32 ± 0.23	0.37 ± 0.20	0.40 ± 0.21
	PT	0.20 ± 0.42	0.50 ± 0.53	0.70 ± 0.48	0.80 ± 0.42	0.80 ± 0.42
Meek's algorithm	RA	0.60 ± 0.10	0.65 ± 0.11	0.68 ± 0.09	0.70 ± 0.07	0.75 ± 0.00
	PA	0.88 ± 0.15	0.85 ± 0.15	0.92 ± 0.13	0.90 ± 0.08	0.93 ± 0.08
	RT	0.03 ± 0.11	0.14 ± 0.19	0.22 ± 0.21	0.25 ± 0.24	0.32 ± 0.20
	PT	0.10 ± 0.32	0.40 ± 0.52	0.60 ± 0.52	0.60 ± 0.52	0.80 ± 0.42

Table 8. Results for the Sachs network (11 nodes and 17 edges).

	size	500	1000	5000	10000	50000
Our algorithm	RA	0.46 ± 0.03	0.52 ± 0.03	0.65 ± 0.02	0.66 ± 0.14	0.83 ± 0.02
	PA	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	RT	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	PT	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Meek's algorithm	RA	0.77 ± 0.05	0.84 ± 0.03	0.91 ± 0.03	0.89 ± 0.02	0.95 ± 0.02
	PA	0.98 ± 0.04	0.99 ± 0.02	1.00 ± 0.00	0.99 ± 0.03	0.99 ± 0.02
	RT	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	PT	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

Table 9. Results for the Child network (20 nodes and 17 edges).

	size	500	1000	5000	10000	50000
Our algorithm	RA	0.38 ± 0.03	0.43 ± 0.02	0.55 ± 0.03	0.59 ± 0.02	0.70 ± 0.02
	PA	0.97 ± 0.06	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	RT	0.00 ± 0.00	0.01 ± 0.04	0.08 ± 0.10	0.37 ± 0.18	0.83 ± 0.24
	PT	0.00 ± 0.00	0.10 ± 0.32	0.40 ± 0.52	1.00 ± 0.00	1.00 ± 0.00
Meek's algorithm	RA	0.72 ± 0.04	0.76 ± 0.05	0.92 ± 0.00	0.92 ± 0.01	0.95 ± 0.02
	PA	0.89 ± 0.07	0.92 ± 0.09	0.90 ± 0.04	0.91 ± 0.04	0.88 ± 0.04
	RT	0.17 ± 0.10	0.38 ± 0.14	0.29 ± 0.13	0.46 ± 0.19	0.48 ± 0.18
	PT	0.90 ± 0.32	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

- Note that the Sachs network has no triplex and hence the 0.00 ± 0.00 in RT and PT scored by both algorithms.
- For the Child network, our algorithm scores better RT for the sample size 50000. Moreover, our algorithm does not score worse PT for the sample sizes 10000 and 50000.
- For the Insurance network, our algorithm scores better RT for the sample sizes 5000 and 50000. Moreover, our algorithm does not score worse PT for any sample size but 500.
- For the Mildew network, our algorithm scores better RT for all the sample sizes. Moreover, our algorithm does not score worse PT for any sample size.
- For the Alarm network, our algorithm scores better RT for the sample sizes 10000 and 50000. Moreover, our algorithm does not score worse PT for any sample size.
- For the Barley network, our algorithm scores better RT for the sample size 50000. Moreover, our algorithm does not score worse PT for any sample size.

Table 10. Results for the Insurance network (27 nodes and 56 edges).

	size	500	1000	5000	10000	50000
Our algorithm	RA	0.11 ± 0.14	0.18 ± 0.16	0.38 ± 0.01	0.40 ± 0.01	0.47 ± 0.01
	PA	0.37 ± 0.48	0.59 ± 0.51	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	RT	0.02 ± 0.03	0.01 ± 0.01	0.20 ± 0.02	0.23 ± 0.00	0.26 ± 0.03
	PT	0.30 ± 0.48	0.60 ± 0.52	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Meek's algorithm	RA	0.16 ± 0.21	0.28 ± 0.24	0.63 ± 0.02	0.69 ± 0.01	0.77 ± 0.01
	PA	0.30 ± 0.39	0.47 ± 0.40	0.82 ± 0.05	0.85 ± 0.03	0.86 ± 0.04
	RT	0.03 ± 0.05	0.05 ± 0.05	0.16 ± 0.06	0.24 ± 0.07	0.22 ± 0.07
	PT	0.40 ± 0.52	0.60 ± 0.52	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

Table 11. Results for the Mildew network (35 nodes and 46 edges).

	size	500	1000	5000	10000	50000
Our algorithm	RA	0.01 ± 0.02	0.09 ± 0.02	0.24 ± 0.02	0.33 ± 0.00	0.46 ± 0.01
	PA	0.81 ± 0.03	0.76 ± 0.06	0.92 ± 0.01	0.94 ± 0.00	0.96 ± 0.00
	RT	0.05 ± 0.01	0.03 ± 0.01	0.08 ± 0.01	0.11 ± 0.00	0.11 ± 0.02
	PT	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Meek's algorithm	RA	0.20 ± 0.03	0.15 ± 0.03	0.27 ± 0.01	0.41 ± 0.00	0.59 ± 0.01
	PA	0.35 ± 0.05	0.26 ± 0.05	0.43 ± 0.03	0.57 ± 0.02	0.59 ± 0.02
	RT	0.03 ± 0.02	0.02 ± 0.01	0.03 ± 0.01	0.06 ± 0.01	0.10 ± 0.02
	PT	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

Table 12. Results for the Alarm network (37 nodes and 46 edges).

	size	500	1000	5000	10000	50000
Our algorithm	RA	0.35 ± 0.03	0.44 ± 0.03	0.57 ± 0.02	0.65 ± 0.02	0.71 ± 0.01
	PA	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
	RT	0.09 ± 0.03	0.11 ± 0.02	0.21 ± 0.05	0.34 ± 0.04	0.38 ± 0.04
	PT	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Meek's algorithm	RA	0.66 ± 0.04	0.76 ± 0.02	0.91 ± 0.03	0.94 ± 0.02	0.96 ± 0.01
	PA	0.82 ± 0.05	0.86 ± 0.03	0.86 ± 0.04	0.83 ± 0.04	0.80 ± 0.02
	RT	0.17 ± 0.04	0.20 ± 0.03	0.31 ± 0.08	0.32 ± 0.12	0.22 ± 0.08
	PT	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

Table 13. Results for the Barley network (48 nodes and 84 edges).

	size	500	1000	5000	10000	50000
Our algorithm	RA	0.00 ± 0.00	0.00 ± 0.00	0.20 ± 0.00	0.21 ± 0.02	0.33 ± 0.04
	PA	0.00 ± 0.00	0.00 ± 0.00	0.83 ± 0.03	0.82 ± 0.01	1.00 ± 0.00
	RT	0.00 ± 0.00	0.00 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	0.07 ± 0.01
	PT	0.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Meek's algorithm	RA	0.00 ± 0.00	0.00 ± 0.00	0.36 ± 0.01	0.38 ± 0.01	0.46 ± 0.01
	PA	0.00 ± 0.00	0.00 ± 0.00	0.65 ± 0.02	0.60 ± 0.01	0.60 ± 0.01
	RT	0.00 ± 0.00	0.00 ± 0.00	0.07 ± 0.01	0.04 ± 0.00	0.05 ± 0.01
	PT	0.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

Since we do not have any reason to deem any of the four performance measures in our experiments (i.e. RA, PA, RT and PT) more important than the others, we can conclude that our algorithm performs relatively well compared to Meek's, although Meek's algorithm is tailored to learning the models sampled in the experiments whereas our algorithm is tailored to learning much more general models. Based on our experiments, we therefore recommend to use Meek's algorithm if it is known that the model sampled is a DAG. However, if it is known that the model sampled is a MAMP CG but it is not known whether it is a DAG, then we recommend to use our algorithm: It guarantees reasonable performance if the model sampled is really a DAG while it accommodates the possibility that the model sampled is a more general MAMP CG. We believe that this is an interesting trade-off.

6. Discussion

MAMP CGs are a recently introduced family of models that is based on graphs that may have undirected, directed and bidirected edges. They unify and generalize AMP and MVR CGs. In this paper, we have presented an algorithm for learning a MAMP CG from a probability distribution p which is faithful to it. We have also proved that the algorithm is correct. The algorithm consists of two phases: The first phase aims at learning adjacencies, whereas the second phase aims at directing some of the adjacencies learnt by applying some rules. It is worth mentioning that, whereas the rules R1, R2 and R4 only involve three or four nodes, the rule R3 may involve more. Unfortunately, we have not succeeded so far in proving the correctness of our algorithm with a simpler R3. Note that the output of our algorithm would be the same. The only benefit might be a decrease in running time. Finally, we have implemented and evaluated our algorithm. The evaluation has shown that our algorithm performs well.

The correctness of our algorithm relies upon the assumption that p is faithful to some MAMP CG. This is a strong requirement that we would like to weaken, e.g. by replacing it with the milder assumption that p satisfies the composition property. Specifically, p satisfies the composition property when $X \perp_p Y | Z \wedge X \perp_p W | Z \Rightarrow X \perp_p Y \cup W | Z$ for all X, Y, Z and W pairwise disjoint subsets of V. Note that if p is a Gaussian distribution, then it satisfies the composition property regardless of whether it is faithful or not to some MAMP CG (Studený, 2005, Corollary 2.4).

When assuming faithfulness is not reasonable, the correctness of a learning algorithm may be redefined as follows. Given a MAMP CG G, we say that p is Markovian with respect to G when $X \perp_p Y | Z$ if $X \perp_G Y | Z$ for all X, Y and Z pairwise disjoint subsets of V. We say that a learning algorithm is correct when it returns a MAMP CG H such that p is Markovian with respect to H and p is not Markovian with respect to any MAMP CG F such that $I(H) \subset I(F)$.

Correct algorithms for learning DAGs and LWF CGs under the composition property assumption exist (Chickering and Meek, 2002; Nielsen et al., 2003; Peña et al., 2014). The way in which these algorithms proceed (i.e. score+search based approach) is rather different from that of the algorithm presented in this paper (i.e. constraint based approach). In a nutshell, they can be seen as consisting of two phases: A first phase that starts from the empty graph H and adds single edges to it until p is Markovian with respect to H, and a second phase that removes single edges from H until p is Markovian with respect to H and p is not Markovian with respect to any graph F such that $I(H) \subset I(F)$. The success of the first phase is guaranteed by the composition property assumption, whereas the success of the second phase is guaranteed by the so-called Meek's conjecture (Meek, 1997). Specifically, given two DAGs F and H such that $I(H) \subseteq I(F)$, Meek's conjecture states that we can transform F into H by a sequence of operations such that, after each operation, F is a DAG and $I(H) \subseteq I(F)$. The operations consist in adding a single edge to F, or replacing F with a triplex equivalent DAG. Meek's conjecture was proven to be true by Chickering (2002, Theorem 4). The extension of Meek's conjecture to LWF CGs was proven to be true by Peña et al. (2014, Theorem 1). The extension of Meek's conjecture to AMP and MVR CGs was proven to be false by Peña (2014a, Example 1) and Sonntag and Peña (2015), respectively. Unfortunately, the extension of Meek's conjecture to MAMP CGs does not hold either, as the following example illustrates.

Example 2. *The MAMP CGs F and H below show that the extension of Meek's conjecture to MAMP CGs does not hold.*

$$
\begin{tikzcd}[column sep=small]
 & A \arrow[d] & B \arrow[d] \\
C \arrow[r, dash] & D \arrow[r, dash] & E \\
 & F &
\end{tikzcd}
\quad
\begin{tikzcd}[column sep=small]
 & A \arrow[d] & B \arrow[dl, dash] \arrow[d, dash] \\
C \arrow[r, dash] & D \arrow[r, dash] & E \\
 & H &
\end{tikzcd}
\quad
\begin{tikzcd}[column sep=small]
 & A \arrow[d] & B \arrow[d, leftrightarrow] \\
C \arrow[r, dash] & D \arrow[r, dash] & E \\
 & F' &
\end{tikzcd}
$$

We can describe $I(F)$ and $I(H)$ by listing all the separators between any pair of distinct nodes. We indicate whether the separators correspond to F or H with a superscript. Specifically,

- $\mathcal{S}^F_{AD} = \mathcal{S}^F_{BE} = \mathcal{S}^F_{CD} = \mathcal{S}^F_{DE} = \varnothing$,
- $\mathcal{S}^F_{AB} = \{\varnothing, \{C\}, \{D\}, \{E\}, \{C, D\}, \{C, E\}\}$,
- $\mathcal{S}^F_{AC} = \{\varnothing, \{B\}, \{E\}, \{B, E\}\}$,
- $\mathcal{S}^F_{AE} = \{\varnothing, \{B\}, \{C\}, \{B, C\}\}$,
- $\mathcal{S}^F_{BC} = \{\varnothing, \{A\}, \{D\}, \{A, D\}, \{A, D, E\}\}$,
- $\mathcal{S}^F_{BD} = \{\varnothing, \{A\}, \{C\}, \{A, C\}\}$, *and*
- $\mathcal{S}^F_{CE} = \{\{A, D\}, \{A, B, D\}\}$.

Likewise,

- $\mathcal{S}^H_{AD} = \mathcal{S}^H_{BD} = \mathcal{S}^H_{BE} = \mathcal{S}^H_{CD} = \mathcal{S}^H_{DE} = \varnothing$,
- $\mathcal{S}^H_{AB} = \{\varnothing, \{C\}, \{E\}, \{C, E\}\}$,
- $\mathcal{S}^H_{AC} = \{\varnothing, \{B\}, \{E\}, \{B, E\}\}$,
- $\mathcal{S}^H_{AE} = \{\varnothing, \{B\}, \{C\}, \{B, C\}\}$,
- $\mathcal{S}^H_{BC} = \{\{A, D\}, \{A, D, E\}\}$, *and*
- $\mathcal{S}^H_{CE} = \{\{A, D\}, \{A, B, D\}\}$.

Then, $I(H) \subseteq I(F)$ because $\mathcal{S}^H_{XY} \subseteq \mathcal{S}^F_{XY}$ for all $X, Y \in \{A, B, C, D, E\}$ with $X \neq Y$. Moreover, the MAMP CG F' above is the only MAMP CG that is triplex equivalent to F, whereas there is no MAMP CG that is triplex equivalent to H. Obviously, one cannot transform F or F' into H by adding a single edge.

While the example above compromises the development of score+search learning algorithms that are correct and efficient under the composition property assumption, it is not clear to us whether it also does it for constraint based algorithms. This is something we plan to study.

Acknowledgments

The first author is supported by the Center for Industrial Information Technology (CENIIT) and a so-called career contract at Linköping University, and by the Swedish Research Council (ref. 2010-4808). The second author is supported by the Spanish Ministry of Economy and Competitiveness under project TIN2013-46638-C3-2-P and by the European Regional Development Fund (FEDER).

Appendix: Proof of Correctness

This appendix is devoted to prove that the algorithm for learning MAMP CGs in Table 1 is correct. We start by proving some auxiliary results.

Lemma 1. *After line 8, 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 $A \not\perp_p B|S$ for all $S \subseteq V \setminus \{A,B\}$ by the faithfulness assumption. Consequently, $A \in ad_H(B)$ at all times. On the other hand, if $A \notin ad_G(B)$, then $A \perp_p B|pa_G(A)$ or $A \perp_p B|ne_G(A) \cup pa_G(A \cup ne_G(A))$ (Peña, 2014b, Theorem 5). Note that, as shown before, $ne_G(A) \cup pa_G(A \cup ne_G(A)) \subseteq [ad_H(A) \cup ad_H(ad_H(A))] \setminus \{A,B\}$ at all times. Therefore, there exists some S in line 5 such that $A \perp_p B|S$ and, thus, the edge $A - B$ will be removed from H in line 7. Consequently, $A \notin ad_H(B)$ after line 8. $\square$

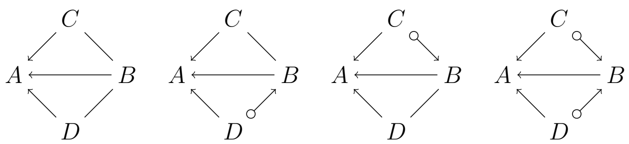
Lemma 2. *The rules R1-R4 block the end of an edge only if the edge is not a directed edge in G pointing in the direction of the block.*

Proof. According to the antecedent of R1, G has a triplex $(\{A,C\},B)$. Then, G has an induced subgraph of the form $A \circ\!\!\rightarrow B \leftarrow\!\!\circ C$, $A \circ\!\!\rightarrow B - C$ or $A - B \leftarrow\!\!\circ C$. In either case, the consequent of R1 holds.

According to the antecedent of R2, (i) G does not have a triplex $(\{A,C\},B)$, (ii) $A \circ\!\!\rightarrow B$ or $A - B$ is in G, (iii) $B \in ad_G(C)$, and (iv) $A \notin ad_G(C)$. Then, $B \rightarrow C$ or $B - C$ is in G. In either case, the consequent of R2 holds.

According to the antecedent of R3, (i) G has a path from A to B with no directed edge pointing in the direction of A, and (ii) $A \in ad_G(B)$. Then, $A \leftarrow B$ cannot be in G because G has no semidirected cycle. Then, the consequent of R3 holds.

According to the antecedent of R4, neither $B \rightarrow C$ nor $B \rightarrow D$ are in G. Assume to the contrary that $A \leftarrow B$ is in G. Then, G must have an induced subgraph of one of the following forms:



because, otherwise, G has a semidirected cycle. However, either case contradicts that $A \in S_{CD}$. $\square$

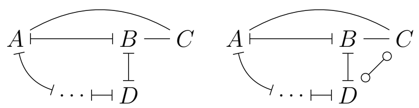
Lemma 3. *After line 11, H has a block at the end of an edge only if the edge is not a directed edge in G pointing in the direction of the block.*

Proof. In Lemma 2, we have proved that any of the rules R1-R4 blocks the end of an edge only if the edge is not a directed edge in G pointing in the direction of the block. Of course, for this to hold, every block in the antecedent of the rule must be on the end of an edge that is not a directed edge in G pointing in the direction of the block. This implies that, after line 9, H has a block at the end of an edge only if the edge is not a directed edge in G pointing in the direction of the block, because H has no blocks before line 9. However, to prove that this result also holds after line 11, we have to prove that line 10 blocks the end of an edge in H only if the edge is not a directed edge in G pointing in the direction of the block. To do so, consider any cycle ρ_H in H that is of length greater than three, chordless, and without blocks. Let ρ_G denote the cycle in G corresponding to the sequence of nodes in ρ_H. Note that no edge in ρ_H can be directed or bidirected in ρ_G because, otherwise, a subroute of the form $A \circ\!\!\rightarrow B \leftarrow\!\!\circ C$ or $A \circ\!\!\rightarrow B - C$ exists in ρ_G since G has no directed cycle. This implies that G contains a triplex $(\{A,C\},B)$ because A and C cannot be adjacent in G since ρ_G is chordless, which implies that $A \vdash\!\!\circ B \circ\!\!\dashv C$ is in H by R1 in line 9, which contradicts that ρ_H has no blocks. Therefore, every edge in ρ_H is undirected in ρ_G and, thus, line 10 blocks the end of an edge in H only if the edge is not a directed edge in G pointing in the direction of the block. $\square$

Lemma 4. *After line 11, H does not have any induced subgraph of the form* $\overparen{A \vdash\!\!\circ B - C}$.

Proof. Assume to the contrary that the lemma does not hold. We interpret the execution of lines 9-11 as a sequence of block addings and, for the rest of the proof, one particular sequence of these block addings is fixed. Fixing this sequence is a crucial point upon which some important later steps of the proof are based. Since there may be several induced subgraphs of H of the form under study after lines 9-11, let us consider any of the induced subgraphs $\widehat{A \vdash\!\!\multimap B - C}$ that appear firstly during execution of lines 9-11 and fix it for the rest of the proof. Now, consider the following cases.

Case 0: Assume that $A \vdash\!\!\dashv B$ is in H due to line 10. Then, after line 10, H had an induced subgraph of one of the following forms, where possible additional edges between C and internal nodes of the route $A \vdash\!\!\dashv \ldots \vdash\!\!\dashv D$ are not shown:



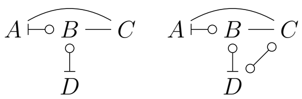
case 0.1 case 0.2

Note that C cannot belong to the route $A \vdash\!\!\multimap \ldots \vdash\!\!\multimap D$ because, otherwise, the cycle $A \vdash\!\!\multimap \ldots \vdash\!\!\multimap D \vdash\!\!\multimap B \multimap A$ would not have been chordless.

Case 0.1: If $B \notin S_{CD}$ then $B \circ\!\!-\!\!\dashv C$ is in H by R1, else $B \vdash\!\!\multimap C$ is in H by R2. Either case is a contradiction.

Case 0.2: Recall from line 10 that the cycle $A \vdash\!\!\dashv \ldots \vdash\!\!\dashv D \vdash\!\!\dashv B \vdash\!\!\dashv A$ is of length greater than three and chordless, which implies that there is no edge between A and D in H. Thus, if $C \notin S_{AD}$ then $A \vdash\!\!\multimap C$ is in H by R1, else $B \circ\!\!-\!\!\dashv C$ is in H by R4. Either case is a contradiction.

Case 1: Assume that $A \vdash\!\!\multimap B$ is in H due to R1. Then, after R1 was applied to $A \circ\!\!-\!\!\circ B$, H had an induced subgraph of one of the following forms:

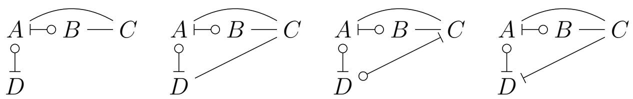


case 1.1 case 1.2

Case 1.1: If $B \notin S_{CD}$ then $B \circ\!\!-\!\!\dashv C$ is in H by R1, else $B \vdash\!\!\multimap C$ is in H by R2. Either case is a contradiction.

Case 1.2: If $C \notin S_{AD}$ then $A \vdash\!\!\multimap C$ is in H by R1, else $B \circ\!\!-\!\!\dashv C$ is in H by R4. Either case is a contradiction.

Case 2: Assume that $A \vdash\!\!\multimap B$ is in H due to R2. Then, after R2 was applied to $A \circ\!\!-\!\!\circ B$, H had an induced subgraph of one of the following forms:



case 2.1 case 2.2 case 2.3 case 2.4

Case 2.1: If $A \notin S_{CD}$ then $A \circ\!\!-\!\!\dashv C$ is in H by R1, else $A \vdash\!\!\multimap C$ is in H by R2. Either case is a contradiction.

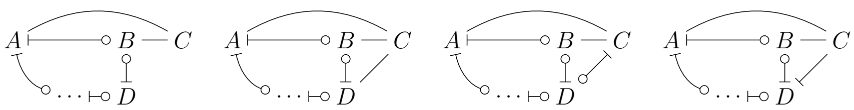
Case 2.2: Note that $\widehat{D \vdash\!\!\multimap A - C}$ cannot be an induced subgraph of H after lines 9-11 because, otherwise, it would contradict the assumption that $\widehat{A \vdash\!\!\multimap B - C}$ is one of the firstly induced subgraph of that form that appeared during the execution of lines 9-11. Then, $A \vdash\!\!\multimap C$, $A \dashv C$, $D \circ\!\!-\!\!\dashv C$ or $D \vdash C$ must be in H after lines 9-11. However, either of the first two cases is a contradiction. The third case can be reduced to Case 2.3 as follows. The fourth case can be reduced to

Case 2.4 similarly. The third case implies that the block at C in $D \circ\!\dashv C$ is added at some moment in the execution of lines 9-11. This moment must happen later than immediately after adding the block at A in $A \vdash\!\circ B$, because immediately after adding this block the situation is the one depicted by the above figure for Case 2.2. Then, when the block at C in $D \circ\!\dashv C$ is added, the situation is the one depicted by the above figure for Case 2.3.

Case 2.3: Assume that the situation of this case occurs at some moment in the execution of lines 9-11. Then, $A \circ\!\dashv C$ is in H by R3 after lines 9-11, which is a contradiction.

Case 2.4: Assume that the situation of this case occurs at some moment in the execution of lines 9-11. If $C \notin S_{BD}$ then $B \vdash\!\circ C$ is in H by R1 after lines 9-11, else $B \circ\!\dashv C$ is in H by R2 after lines 9-11. Either case is a contradiction.

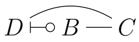
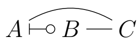
Case 3: Assume that $A \vdash\!\circ B$ is in H due to R3. Then, after R3 was applied to $A \circ\!\!-\!\!\circ B$, H had a subgraph of one of the following forms, where possible additional edges between C and internal nodes of the route $A \vdash\!\circ \ldots \vdash\!\circ D$ are not shown:



case 3.1 case 3.2 case 3.3 case 3.4

Note that C cannot belong to the route $A \vdash\!\circ \ldots \vdash\!\circ D$ because, otherwise, R3 could not have been applied since the cycle $A \vdash\!\circ \ldots \vdash\!\circ D \vdash\!\circ B \multimap A$ would not have been chordless.

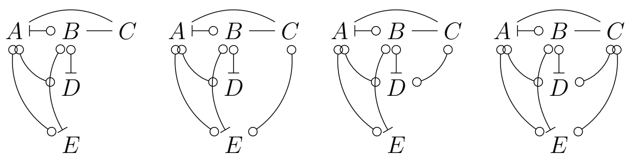
Case 3.1: If $B \notin S_{CD}$ then $B \circ\!\dashv C$ is in H by R1, else $B \vdash\!\circ C$ is in H by R2. Either case is a contradiction.

Case 3.2: Note that  cannot be an induced subgraph of H after lines 9-11 because, otherwise, it would contradict the assumption that  is one of the firstly induced subgraph of that form that appeared during the execution of lines 9-11. Then, $B \vdash\!\circ C$, $B \dashv C$, $D \circ\!\dashv C$ or $D \vdash C$ must be in H after lines 9-11. However, either of the first two cases is a contradiction. The third case can be reduced to Case 3.3 as follows. The fourth case can be reduced to Case 3.4 similarly. The third case implies that the block at C in $D \circ\!\dashv C$ is added at some moment in the execution of lines 9-11. This moment must happen later than immediately after adding the block at A in $A \vdash\!\circ B$, because immediately after adding this block the situation is the one depicted by the above figure for Case 3.2. Then, when the block at C in $D \circ\!\dashv C$ is added, the situation is the one depicted by the above figure for Case 3.3.

Case 3.3: Assume that the situation of this case occurs at some moment in the execution of lines 9-11. Then, $B \circ\!\dashv C$ is in H by R3 after lines 9-11, which is a contradiction.

Case 3.4: Assume that the situation of this case occurs at some moment in the execution of lines 9-11. Note that C is not adjacent to any node of the route $A \vdash\!\circ \ldots \vdash\!\circ D$ besides A and D. To see it, assume to the contrary that C is adjacent to some nodes $E_1, \ldots, E_n \neq A, D$ of the route $A \vdash\!\circ \ldots \vdash\!\circ D$. Assume without loss of generality that E_i is closer to A in the route than E_{i+1} for all $1 \leq i < n$. Now, note that $E_n \vdash\!\circ C$ must be in H by R3 after lines 9-11. This implies that $E_{n-1} \vdash\!\circ C$ must be in H by R3 after lines 9-11. By repeated application of this argument, we can conclude that $E_1 \vdash\!\circ C$ must be in H after lines 9-11 and, thus, $A \vdash\!\circ C$ must be in H by R3 after lines 9-11, which is a contradiction. Therefore, if C is not adjacent to any node of the route $A \vdash\!\circ \ldots \vdash\!\circ D$ besides A and D, then the cycle $A \vdash\!\circ \dots \vdash\!\circ D \vdash C - A$ is chordless and, thus, $A \vdash\!\circ C$ must be in H by R3 after lines 9-11, which is a contradiction.

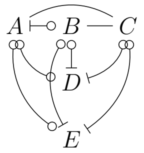
Case 4: Assume that $A \vdash\!\circ B$ is in H due to R4. Then, after R4 was applied to $A \circ\!\!-\!\!\circ B$, H had an induced subgraph of one of the following forms:



case 4.1 case 4.2 case 4.3 case 4.4

Cases 4.1-4.3: If $B \notin S_{CD}$ or $B \notin S_{CE}$ then $B \circ\!\!\dashv C$ is in H by R1, else $B \vdash\!\circ C$ is in H by R2. Either case is a contradiction.

Case 4.4: Assume that $C \in S_{DE}$. Then, $B \circ\!\!\dashv C$ is in H by R4, which is a contradiction. On the other hand, assume that $C \notin S_{DE}$. Then, it follows from applying R1 that H has an induced subgraph of the form

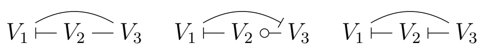


Note that $A \in S_{DE}$ because, otherwise, R4 would not have been applied. Then, $A \vdash\!\circ C$ is in H by R4, which is a contradiction.

□

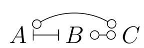
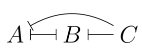
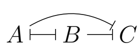
Lemma 5. *After line 11, every chordless cycle $\rho : V_1, \dots, V_n = V_1$ in H that has an edge $V_i \vdash V_{i+1}$ also has an edge $V_j \dashv V_{j+1}$.*

Proof. Assume for a contradiction that ρ is of the length three such that $V_1 \vdash V_2$ occur and neither $V_2 \dashv V_3$ nor $V_1 \vdash V_3$ occur. Note that $V_2 \vdash\!\dashv V_3$ cannot occur either because, otherwise, $V_1 \vdash\!\circ V_3$ must occur by R3. Since $V_1 \vdash V_3$ contradicts the assumption, then $V_1 \vdash\!\dashv V_3$ must occur. However, this implies that $V_1 \vdash\!\dashv V_2$ must occur by R3, which contradicts the assumption. Similarly, $V_1 \vdash\!\dashv V_3$ cannot occur either. Then, ρ is of one of the following forms:



The first form is impossible by Lemma 4. The second form is impossible because, otherwise, $V_2 \circ\!\!\dashv V_3$ would occur by R3. The third form is impossible because, otherwise, $V_1 \vdash V_3$ would be occur by R3. Thus, the lemma holds for cycles of length three.

Assume for a contradiction that ρ is of length greater than three and has an edge $V_i \vdash V_{i+1}$ but no edge $V_j \dashv V_{j+1}$. Note that if $V_l \vdash\!\circ V_{l+1} \circ\!\!-\!\!\circ V_{l+2}$ is a subroute of ρ, then either $V_{l+1} \vdash\!\circ V_{l+2}$ or $V_{l+1} \dashv V_{l+2}$ is in ρ by R1 and R2. Since ρ has no edge $V_j \dashv V_{j+1}$, $V_{l+1} \vdash\!\circ V_{l+2}$ is in ρ. By repeated application of this reasoning together with the fact that ρ has an edge $V_i \vdash V_{i+1}$, we can conclude that every edge in ρ is $V_k \vdash\!\circ V_{k+1}$. Then, by repeated application of R3, observe that every edge in ρ is $V_k \vdash\!\dashv V_{k+1}$, which contradicts the assumption. □

Lemma 6. *If H has an induced subgraph of the form*  *after line 11, then the induced subgraph must actually be of the form*  ,  *or*  .

Proof. Lemmas 4 and 5 together with R3 rule out any other possibility. □

Lemma 7. *All the undirected edges in H at line 18 that are of the form $\vdash\!\dashv$ after line 11 are undirected edges in G.*

Proof. The undirected edges in H at line 18 that are of the form $\vdash\!\dashv$ after line 11 are those added to H in lines 14 and 16. We first prove that the undirected edges added to H in line 14 are undirected edges in G. Any undirected edges $A - B$ and $B - C$ added to H in line 14 imply that H has an induced subgraph $A \vdash\!\dashv B \vdash\!\dashv C$ with $B \in S_{AC}$ after line 11, which implies that (i) A and B as well as B and C are adjacent in G whereas A and C are not adjacent in G by Lemma 1, and (ii) G has no directed edge between A and B or B and C by Lemma 3. Then, $A - B - C$ must be in G because $B \in S_{AC}$.

We now prove that the undirected edges added to H in line 16 are undirected edges in G. As shown in the paragraph above, the result holds after having executed line 16 zero times. Assume as induction hypothesis that the result also holds after having executed line 16 n times. When line 16 is executed for the $(n+1)$-th time, H has an induced subgraph of the form $A \leftrightarrow B - C$. This implies that H has an induced subgraph of the form $A \vdash\!\dashv B \circ\!\!-\!\!\circ C$ after line 11, which implies that the induced subgraph is actually of the form $A \vdash\!\dashv B \vdash\!\dashv C$ by Lemma 6. Then, the undirected edges $B - C$ and $A - C$ have been added to H in previous executions of lines 14 and 16. Then, $B - C$ and $A - C$ are in G by the induction hypothesis and, thus, $A - B$ must be in G too due to the constraint C2. Consequently, the desired result holds after having executed line 16 $n + 1$ times. $\square$

Lemma 8. *At line 18, G and H have the same triplexes.*

Proof. We first prove that any triplex in H at line 18 is in G. Assume to the contrary that H at line 18 has a triplex $(\{A, C\}, B)$ that is not in G. This is possible if and only if H has an induced subgraph of one of the following forms after line 11:

$$A - B \circ\!\!\dashv C \qquad A \vdash B - C \qquad A \vdash B \circ\!\!\dashv C \qquad A \vdash\!\dashv B \multimap C \qquad A \vdash\!\dashv B \vdash\!\dashv C$$

Note that the induced subgraphs above together with Lemma 1 imply that A is adjacent to B in G, B is adjacent to C in G, and A is not adjacent to C in G. This together with the assumption made above that G has no triplex $(\{A, C\}, B)$ implies that $B \in S_{AC}$. Now, note that the first and third induced subgraphs above are impossible because, otherwise, $A \circ\!\!\dashv B$ would be in H by R2. Likewise, the second and fourth induced subgraphs above are impossible because, otherwise, $B \vdash\!\!\circ C$ would be in H by R2. Now, note that any triplex that is added to H in line 13 due to the fifth induced subgraph above is removed from H in line 14 because, as shown above, $B \in S_{AC}$. Finally, note that no triplex is added to H in lines 15-17.

We now prove that any triplex $(\{A, C\}, B)$ in G is in H at line 18. Note that $B \notin S_{AC}$. Consider the following cases.

Case 1: Assume that the triplex in G is of the form $A \to B \circ\!\!-\!\!\circ C$ (respectively $A \circ\!\!-\!\!\circ B \leftarrow C$). Then, after line 11, $A \vdash B \circ\!\!\dashv C$ (respectively $A \vdash\!\!\circ B \dashv C$) is in H by Lemmas 1 and 3. Then, the triplex is added to H in lines 12-13. Moreover, the triplex added is of the form $A \to B \circ\!\!-\!\!\circ C$ (respectively $A \circ\!\!-\!\!\circ B \leftarrow C$) and, thus, it does not get removed from H in lines 14-17, because all these lines do is replacing bidirected edges in H with undirected edges.

Case 2: Assume that the triplex in G is of the form $A \leftrightarrow B \circ\!\!-\!\!\circ C$ or $A \circ\!\!-\!\!\circ B \leftrightarrow C$. Then, after line 11, $A \vdash\!\!\circ B \circ\!\!\dashv C$ is in H by Lemmas 1 and 3. Then, the triplex is added to H in lines 12-13. Moreover, the triplex cannot get removed from H in lines 14-17. To see it, consider the following cases.

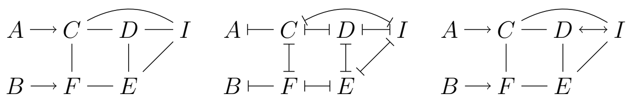
Case 2.1: Assume that the subgraph $A \vdash\!\!\circ B \circ\!\!\dashv C$ in H after line 11 is actually of the form $A \vdash B \circ\!\!\dashv C$ or $A \vdash\!\!\circ B \dashv C$. Then, the triplex does not get removed from H in lines 14-17, because all these lines do is replacing bidirected edges in H with undirected edges.

Case 2.2: Assume that the subgraph $A \mapstochar\!\!\multimap B \multimapinv\!\!\!\dashv C$ in H after line 11 is actually of the form $A \mapsto\!\!\!\dashv B \mapsto\!\!\!\dashv C$. Note that all lines 14-17 do is replacing bidirected edges in H with undirected edges. Therefore, the triplex gets removed from H only if $A - B - C$ is in H at line 18. However, this implies that $A - B - C$ is in G by Lemma 7, which is a contradiction.

□

It is worth noting that one may think that Lemma 4 implies that H does not have any induced subgraph of the form $A \leftrightarrow B - C$ after line 14 and, thus, that lines 15-17 are not needed. However, this is wrong as the following example illustrates.

Example 3. *The MAMP CG G below shows that lines 15-17 are necessary.*



G *H after line 11*   *H after line 14*

We are now ready prove the correctness of our algorithm.

Theorem 1. *At line 18, H is a MAMP CG that is Markov equivalent to G.*

Proof. First, note that Lemmas 1 and 8 imply that H at line 18 has the same adjacencies and triplexes as G.

Now, we show that H at line 18 satisfies the constraint C1. Lemma 5 implies that H has no semidirected chordless cycle after line 13. This implies that H has no semidirected chordless cycle at line 18, because all lines 14-17 do is replacing bidirected edges in H with undirected edges. To see that this in turn implies that H has no semidirected cycle at line 18, assume to the contrary that H has no semidirected chordless cycle but it has a semidirected cycle $\rho : V_1, \ldots, V_n = V_1$ with a chord between V_i and V_j with $i < j$. Then, divide ρ into the cycles $\rho_L : V_1, \ldots, V_i, V_j, \ldots, V_n = V_1$ and $\rho_R : V_i, \ldots, V_j, V_i$. Note that ρ_L or ρ_R is a semidirected cycle. Then, H has a semidirected cycle that is shorter than ρ. By repeated application of this reasoning, we can conclude that H has a semidirected chordless cycle, which is a contradiction.

Now, we show that H at line 18 satisfies the constraint C2. Assume to the contrary that H has a cycle $\rho : V_1, \ldots, V_n = V_1$ such that $V_1 \leftrightarrow V_2$ is in H and $V_i - V_{i+1}$ is in H for all $1 < i < n$. Note that ρ must be of length greater than three by lines 15-17, i.e. $n > 3$. Note also that that $V_1 \leftrightarrow V_2$ is in H at line 18 implies that $V_1 \mapsto\!\!\!\dashv V_2$ is in H at line 11. Consider the following cases.

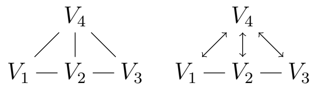
Case 1: Assume that V_1 and V_3 are not adjacent in H at line 18. Then, $V_2 \mapstochar\!\!\multimap V_3$ or $V_2 \multimapinv\!\!\!\dashv V_3$ is in H after line 11 by R1 or R2. In fact, $V_2 \mapsto\!\!\!\dashv V_3$ must be in H after line 11 because, otherwise, $V_2 \to V_3$ or $V_2 \leftarrow V_3$ is in H at line 18, which is a contradiction.

Case 2: Assume that V_1 and V_3 are adjacent in H at line 18. That $V_2 - V_3$ is in H at line 18 implies that $V_1 \mapsto\!\!\!\dashv V_2 \mapsto\!\!\!\dashv V_3$ or $V_1 \mapsto\!\!\!\dashv V_2 - V_3$ is in H at line 11. In fact, $V_2 \mapsto\!\!\!\dashv V_3$ must be in H after line 11 by Lemma 6.

In either case above, $V_2 \mapsto\!\!\!\dashv V_3$ is in H after line 11 and, thus, $V_2 - V_3$ is in G by Lemma 7. By repeated application of this argument, we can conclude that $V_i - V_{i+1}$ is in G for all $2 < i < n$, which implies that $V_1 - V_2$ is also in G by the constraints C1 and C2. This implies that V_1 and V_3 are adjacent in G because, otherwise, G and H have not the same triplexes, which contradicts Lemma 8. Then, V_1 and V_3 are adjacent in H by Lemma 1. In fact, $V_1 \leftrightarrow V_3$ must be in H because, otherwise, H has a cycle of length three that violates the constraint C1 or C2 which, as shown above, is a contradiction. Then, H has a cycle that violates the constraint

C2 and that is shorter than ρ, namely $V_1, V_3, \dots, V_n = V_1$. By repeated application of this reasoning, we can conclude that H has a cycle of length three that violates the constraint C2 which, as shown above, is a contradiction.

Finally, we show that H at line 18 satisfies the constraint C3. Assume to the contrary that, at line 18, $V_1 - V_2 - V_3$ is in H, $V_2 \leftrightarrow V_4$ is in H, but $V_1 - V_3$ is not in H. We show below that G (respectively H at line 18) has the graph to the left (respectively right) below as an induced subgraph.



That $V_1 - V_2 - V_3$ is in H at line 18 but $V_1 - V_3$ is not implies that V_1 and V_3 cannot be adjacent in H because, otherwise, H violates the constraint C1 or C2 which, as shown above, is a contradiction. This implies that V_1 and V_3 are not adjacent in G either by Lemma 1. Consider the following cases.

Case 1: Assume that V_1 and V_4 are not adjacent in H at line 18. That $V_2 \leftrightarrow V_4$ is in H at line 18 implies that $V_2 \vdash\!\dashv V_4$ is in H after line 11 and, thus, that $V_1 \vdash\!\circ V_2$ or $V_1 \circ\!\dashv V_2$ is in H after line 11 by R1 or R2. In fact, $V_1 \vdash\!\dashv V_2$ must be in H after line 11 because, otherwise, $V_1 \to V_2$ or $V_1 \leftarrow V_2$ is in H at line 18, which is a contradiction. Then, $V_1 - V_2$ is in G by Lemma 7.

Case 2: Assume that V_1 and V_4 are adjacent in H at line 18. That $V_1 - V_2 \leftrightarrow V_4$ is in H at line 18 implies that $V_1 \vdash\!\dashv V_2 \vdash\!\dashv V_4$ or $V_1 - V_2 \vdash\!\dashv V_4$ is in H after line 11. In fact, $V_1 \vdash\!\dashv V_2$ must be in H after line 11 by Lemma 6. Then, $V_1 - V_2$ is in G by Lemma 7.

In either case above, $V_1 - V_2$ is in G. Likewise, $V_2 - V_3$ is in G. That $V_2 \leftrightarrow V_4$ is in H at line 18 implies that $V_2 \vdash\!\dashv V_4$ is in H after line 11, which implies that $V_2 - V_4$ or $V_2 \leftrightarrow V_4$ is in G by Lemmas 1 and 3. In fact, $V_2 - V_4$ must be in G because, otherwise, G violates the constraint C3 since, as shown above, $V_1 - V_2 - V_3$ is in G but $V_1 - V_3$ is not. Finally, note that V_1 and V_4 as well as V_3 and V_4 must be adjacent in G and H because, otherwise, H at line 18 does not have the same triplexes as G, which contradicts Lemma 8. Specifically, $V_1 - V_4 - V_3$ must be in G and $V_1 \leftrightarrow V_4 \leftrightarrow V_3$ must be in H because, otherwise, G or H violates the constraint C1 or C2 which, as shown above, is a contradiction.

However, that G (respectively H at line 18) has the graph to the left (respectively right) above as an induced subgraph implies that H has a triplex $(\{V_1, V_3\}, V_4)$ that G has not, which contradicts Lemma 8. Then, V_1 and V_3 must be adjacent in H which, as shown above, is a contradiction. □

Theorem 2. *At line 18, H is the EG of the Markov equivalence class of G.*

Proof. Let K denote the graph that contains all and only the edges in H at line 18 that have a block in H after line 11, and let U denote the graph that contains the rest of the edges in H at line 18. Note that every edge in K is undirected, directed or bidirected, whereas every edge in U is undirected. Note also that the edges in U correspond to the edges without blocks in H after line 11. Therefore, U has no cycle of length greater than three that is chordless by line 10. In other words, U is chordal. Then, we can orient all the edges in U without creating triplexes nor directed cycles by using, for instance, the maximum cardinality search algorithm (Koller and Friedman, 2009, p. 312). Consider any such orientation of the edges in U and denote it D. Now, add all the edges in D to K. We show below that K is a MAMP CG that is triplex equivalent to H.

First, we show that K is triplex equivalent to H. Assume the contrary. Clearly, K, H and G have the same adjacencies by Theorem 1. Then, consider the following cases.

Case 1: Assume that K has a triplex $(\{A, C\}, B)$ that is not in H. Then, the triplex is not in G either by Theorem 1 and, thus, $B \in S_{AC}$. Moreover, recall from above that D

has no triplex. Then, both edges in the triplex $(\{A,C\},B)$ cannot be in D. Note also that both edges in the triplex $(\{A,C\},B)$ cannot be outside D because, otherwise, the triplex is in H. Then, the triplex $(\{A,C\},B)$ has one edge in D, say the one between A and B, and the other edge outside D. This implies that $A - B \circ\!\!\dashv C$ is in H after line 11, which implies that $A \circ\!\!\dashv B$ is in H after line 11 by R2 because, as shown above, $B \in S_{AC}$. This is a contradiction.

Case 2: Assume that H has a triplex $(\{A,C\},B)$ that is not in K. Then, the triplex is also in G by Theorem 1 and, thus, $B \notin S_{AC}$. Moreover, recall from above that D has no triplex. Then, both edges in the triplex $(\{A,C\},B)$ cannot be in U. Note also that both edges in the triplex $(\{A,C\},B)$ cannot be outside U because, otherwise, the triplex is in K. Then, the triplex $(\{A,C\},B)$ has one edge in U, say the one between A and B, and the other edge outside D. This implies that $A - B \circ\!\!\dashv C$ is in H after line 11, which implies that $A \vdash\!\!\circ B$ is in H after line 11 by R1 because, as shown above, $B \notin S_{AC}$. This is a contradiction.

Now, we show that K satisfies the constraint C1. Assume to the contrary that there is a semidirected cycle ρ in K. Note that ρ must include some edges that are not in D because, as defined above, D is a DAG. Then, ρ includes some edges that are of the form $\vdash$ or $\vdash\!\!\dashv$ after line 11. Consider the following cases.

Case 1: Assume that ρ includes an edge of the form $\vdash$ after line 11. Then, ρ cannot be chordless by Lemma 5 because, otherwise, ρ includes an edge of the form $\dashv$, which contradicts that ρ is a semidirected cycle by line 12. Then, let $\rho : V_1, \ldots, V_n = V_1$ have a chord between V_i and V_j with $i < j$. Then, divide ρ into the cycles $\rho_L : V_1, \ldots, V_i, V_j, \ldots, V_n = V_1$ and $\rho_R : V_i, \ldots, V_j, V_i$. Note that ρ_L or ρ_R is a semidirected cycle. Then, H has a semidirected cycle that is shorter than ρ. By repeated application of this reasoning, we can conclude that H has a semidirected chordless cycle, which contradicts Lemma 5.

Case 2: Assume that ρ includes an edge of the form $\vdash\!\!\dashv$ and no edge of the form $\vdash$ after line 11. Note that ρ must include an edge of D because, otherwise, all the edges in ρ are undirected or bidirected, which is a contradiction. Then, H has a subgraph of the form $A - B \vdash\!\!\dashv C$ after line 11. Moreover, A and C must be adjacent in H because, otherwise, $A \vdash\!\!\circ B$ or $A \circ\!\!\dashv B$ is in H after line 11 by R1 or R2, which contradicts that $A - B$ is in H after line 11. That $A - B \vdash\!\!\dashv C$ is in H after line 11 together with the fact that A and C are adjacent in H contradict Lemma 6.

Now, we show that K satisfies the constraint C2. Note that the only difference between K and H is that the former may contain a directed edge where the latter has an undirected edge. Therefore, K cannot have other cycles that violate the constraint C2 beside those H has, which are none.

Finally, we show that K satisfies the constraint C3. Assume to the contrary that $V_1 - V_2 - V_3$ is in K, $V_2 \leftrightarrow V_4$ is in K, but $V_1 - V_3$ is not in K. We show below that H at line 18 (respectively K) has the graph to the left (respectively right) below as an induced subgraph.

$$\begin{tikzcd}
 & V_4 \arrow[dl, dash] \arrow[d, dash] \arrow[dr, dash] & \\
V_1 \arrow[r, dash] & V_2 \arrow[r, dash] & V_3
\end{tikzcd}
\qquad
\begin{tikzcd}
 & V_4 \arrow[dl, leftrightarrow] \arrow[d, leftrightarrow] \arrow[dr, leftrightarrow] & \\
V_1 \arrow[r, dash] & V_2 \arrow[r, dash] & V_3
\end{tikzcd}$$

That $V_1 - V_2 - V_3$ is in K but $V_1 - V_3$ is not implies that V_1 and V_3 cannot be adjacent in K because, otherwise, K violates the constraint C1 or C2 which, as shown above, is a contradiction. This implies that V_1 and V_3 are not adjacent in H at line 18 either, because H and K have the same adjacencies. Consider the following cases.

Case 1: Assume that V_1 and V_4 are not adjacent in K. That $V_2 \leftrightarrow V_4$ is in K implies that $V_2 \vdash\!\!\dashv V_4$ is in H after line 11 and, thus, that $V_1 \vdash\!\!\circ V_2$ or $V_1 \circ\!\!\dashv V_2$ is in H after line 11 by R1 or R2. In fact, $V_1 \vdash\!\!\dashv V_2$ must be in H after line 11 because, otherwise,

$V_1 \to V_2$ or $V_1 \leftarrow V_2$ is in H at line 18 and thus in K, which is a contradiction. Then, $V_1 - V_2$ is in H at line 18 by definition of K.

Case 2: Assume that V_1 and V_4 are adjacent in K. That $V_1 - V_2 \leftrightarrow V_4$ is in K implies that $V_1 \mapsto V_2 \mapsto V_4$ or $V_1 - V_2 \mapsto V_4$ is in H after line 11. In fact, $V_1 \mapsto V_2$ must be in H after line 11 by Lemma 6. Then, $V_1 - V_2$ is in H at line 18 by definition of K.

In either case above, $V_1 - V_2$ is in H at line 18. Likewise, $V_2 - V_3$ is in H at line 18. That $V_2 \leftrightarrow V_4$ is in K implies that $V_2 \mapsto V_4$ is in H after line 11, which implies that $V_2 - V_4$ or $V_2 \leftrightarrow V_4$ is in H at line 18. In fact, $V_2 - V_4$ must be in H because, otherwise, H violates the constraint C3 since, as shown above, $V_1 - V_2 - V_3$ is in H but $V_1 - V_3$ is not. Finally, note that V_1 and V_4 as well as V_3 and V_4 must be adjacent in H at line 18 and K because, otherwise, K does not have the same triplexes as H which, as shown above, is a contradiction. Specifically, $V_1 - V_4 - V_3$ must be in H and $V_1 \leftrightarrow V_4 \leftrightarrow V_3$ must be in K because, otherwise, H or K violates the constraint C1 or C2 which, as shown above, is a contradiction.

However, that H at line 18 (respectively K) has the graph to the left (respectively right) above as an induced subgraph implies that K has a triplex $(\{V_1, V_3\}, V_4)$ that H has not which, as shown above, is a contradiction. Then, V_1 and V_3 must be adjacent in K which, as shown above, is a contradiction.

Consequently, K is a MAMP CG that is triplex equivalent to H and, thus, to G by Theorem 1. Now, let us recall how the maximum cardinality search algorithm works. It first unmarks all the nodes in U and, then, iterates through the following step until all the nodes are marked: Select any of the unmarked nodes with the largest number of marked neighbors and mark it. Finally, the algorithm orients every edge in U away from the node that was marked earlier. Clearly, any node can get marked firstly by the algorithm because there is a tie among all the nodes in the first iteration, which implies that either orientation of any edge can occur in D and thus in K. Therefore, either orientation of every edge of U occurs in a MAMP CG K that is triplex equivalent to G. Consequently, the edges in U are in the EG of the Markov equivalence class of G. Recall that the edges in U are all undirected but that they may not be all the undirected edges in H. Specifically, any edge $A - B$ in H after line 18 that is not in U corresponds to an edge $A \mapsto B$ in H after line 11. This implies that $A - B$ is in the EG of the Markov equivalence class of G, because Lemma 7 remains valid if we replace in it G with any other member of the Markov equivalence class of G. Likewise, any edge $A \to B$ (respectively $A \leftrightarrow B$) in H after line 18 corresponds to an edge $A \vdash B$ (respectively $A \mapsto B$) in H after line 11. This implies that $A \to B$ (respectively $A \leftrightarrow B$) is in the EG of the Markov equivalence class of G because $A \to B$ (respectively $A \leftrightarrow B$) is in a member of the class, namely H, and $A \leftarrow B$ (respectively $A \to B$ and $A \leftarrow B$) is in no member of the class, since Lemma 3 remains valid if we replace in it G with any other member of the Markov equivalence class of G. Consequently, the EG of the Markov equivalence class of G has the same edges as H. □

References

Andersson, S. A., Madigan, D. and Perlman, M. D. A Characterization of Markov Equivalence Classes for Acyclic Digraphs. *The Annals of Statistics*, 25:505-541, 1997.

Andersson, S. A., Madigan, D. and Perlman, M. D. Alternative Markov Properties for Chain Graphs. *Scandinavian Journal of Statistics*, 28:33-85, 2001.

Andersson, S. A. and Perlman, M. D. Characterizing Markov Equivalent Classes for AMP Chain Graph Models. *The Annals of Statistics*, 34:939-972, 2006.

Cano, A., Gómez-Olmedo, M. and Moral, S. A Score Based Ranking of the Edges for the PC Algorithm. In *Proceedings of the 4th European Workshop on Probabilistic Graphical Models*, 41-48, 2008.

Chickering, D. M. Optimal Structure Identification with Greedy Search. *Journal of Machine Learning Research*, 3:507-554, 2002.

Chickering, D. M. and Meek, C. Finding Optimal Bayesian Networks. In *Proceedings of 18th Conference on Uncertainty in Artificial Intelligence*, 94-102, 2002.

Colombo, D. and Maathuis, M. H. Order-Independent Constraint-Based Causal Structure Learning. *Journal of Machine Learning Research*, to appear.

Cox, D. R. and Wermuth, N. *Multivariate Dependencies - Models, Analysis and Interpretation*. Chapman & Hall, 1996.

Koller, D. and Friedman, N. *Probabilistic Graphical Models*. MIT Press, 2009.

Koster, J. T. A. Marginalizing and Conditioning in Graphical Models. *Bernoulli*, 8:817-840, 2002.

Lauritzen, S. L. *Graphical Models*. Oxford University Press, 1996.

Ma, Z., Xie, X. and Geng, Z. Structural Learning of Chain Graphs via Decomposition. *Journal of Machine Learning Research*, 9:2847-2880, 2008.

Meek, C. Causal Inference and Causal Explanation with Background Knowledge. In *Proceedings of 11th Conference on Uncertainty in Artificial Intelligence*, 403-418, 1995.

Meek, C. *Graphical Models: Selecting Causal and Statistical Models*. PhD thesis, Carnegie Mellon University, 1997.

Nielsen, J. D., Kočka, T. and Peña, J. M. On Local Optima in Learning Bayesian Networks. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, 435-442, 2003.

Peña, J. M. Learning AMP Chain Graphs and some Marginal Models Thereof under Faithfulness. *International Journal of Approximate Reasoning*, 55:1011-1021, 2014.

Peña, J. M. Marginal AMP Chain Graphs. *International Journal of Approximate Reasoning*, 55:1185-1206, 2014.

Peña, J. M. Learning Marginal AMP Chain Graphs under Faithfulness. In *Proceedings of the 7th European Workshop on Probabilistic Graphical Models*, 382-395.

Peña, J. M., Sonntag, D. and Nielsen, J. D. An Inclusion Optimal Algorithm for Chain Graph Structure Learning. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, 778-786, 2014.

Ramsey, J., Spirtes, P. and Zhang, J. Adjacency-Faithfulness and Conservative Causal Inference. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, 401-408, 2006.

Richardson, T. and Spirtes, P. Ancestral Graph Markov Models. *The Annals of Statistics*, 30:962-1030, 2002.

Sadeghi, K. Stable Mixed Graphs. *Bernoulli*, 19:2330-2358, 2013.

Sadeghi, K. and Lauritzen, S. L. Markov Properties for Mixed Graphs. *Bernoulli*, 20:676-696, 2014.

Sonntag, D. and Peña, J. M. Learning Multivariate Regression Chain Graphs under Faithfulness. In *Proceedings of the 6th European Workshop on Probabilistic Graphical Models*, 299-306, 2012.

Sonntag, D. and Peña, J. M. Chain Graph Interpretations and their Relations Revisited. *International Journal of Approximate Reasoning*, to appear.

Sonntag, D., Peña, J. M. and Gómez-Olmedo, M. Approximate Counting of Graphical Models Via MCMC Revisited. *International Journal of Intelligent Systems*, to appear.

Spirtes, P., Glymour, C. and Scheines, R. *Causation, Prediction, and Search*. Springer-Verlag, 1993.

Studený, M. A Recovery Algorithm for Chain Graphs. *International Journal of Approximate Reasoning*, 17:265-293, 1997.

Studený, M. *Probabilistic Conditional Independence Structures*. Springer, 2005.