Greedy Relaxations of the Sparsest Permutation Algorithm

Abstract

There has been an increasing interest in methods that exploit permutation reasoning to search for directed acyclic causal models, including the "Ordering Search" of Teyssier and Kohler and GSP of Solus, Wang and Uhler. We extend the methods of the latter by a permutation-based operation *tuck*, and develop a class of algorithms, namely GRaSP, that are efficient and pointwise consistent under increasingly weaker assumptions than faithfulness. The most relaxed form of GRaSP outperforms many state-of-the-art causal search algorithms in simulation, allowing efficient and accurate search even for dense graphs and graphs with more than 100 variables.

1 INTRODUCTION

Searching for causal models by identifying patterns of conditional independence in observational data has become a well-established activity, though it is not without detractors. For one thing, it is commonly believed that the only correct method for establishing causal relationships is through experimental manipulation, as is done in a randomized controlled trial. Accordingly, causal inference from observational data alone can be seen as second-rate. This is not completely unreasonable; many causal search algorithms, even in ideal conditions, demonstrate poor performance, calling into question whether their inferences can be relied upon. Furthermore, the theoretical assumptions made by these algorithms are often criticized for being too strong. More specifically, these algorithms customarily assume that the true model belongs to a model class with no latent variables or no cycles, and that the patterns of conditional independence in the data generating distribution can be represented by the assumed model class exactly. The latter assumption is called the causal faithfulness condition and can be violated by unexpected patterns of conditional independence that arise from subtleties in the distribution, such as determinism or path cancellation.

The most common model class assumed by causal search algorithms is characterized by directed acyclic graphs (DAGs). Many algorithms for causal inference search the space of causal DAGs, such as the PC ("Peter and Clark", [Spirtes et al., 2000]) and GES ("Greedy equivalence Search", [Chickering, 2002]) algorithms, and provably return a set of DAGs that contains the true model under the causal faithfulness condition. However, the performance of these algorithms tends to fall short of their theoretical claims, especially when the true model is densely connected. One hypothesis that explains this phenomenon is that almost-violations of the causal faithfulness condition frequently occur and disrupt the search procedures [Uhler et al., 2013]. Accordingly, the performance of these algorithms might be improved by relaxing the causal faithfulness condition, as is done by the SP ("Sparsest Permutation") algorithm of Raskutti and Uhler [2018]. SP considers the space of variable orderings and builds a DAG using a procedure inspired by Verma and Pearl [1988], where the parents of each variable are selected from the preceding variables in the permutation. Ultimately, the permutations that induce DAGs with the minimal edge count are selected.

Raskutti and Uhler [2018] proved that if the data generating distribution is a graphoid, then the set of DAGs returned by SP contains the true model under an assumption strictly weaker than the causal faithfulness condition. While SP can be very accurate and sensibly interpreted, it is super exponential in the number of variables, that is, if there are n variables, then there are $n!$ permutations that must be visited. In practice, it is limited to a maximum of about nine variables due to its computational complexity. This naturally begs the question: is there an algorithm that is equally accurate in most cases for such data, but that can scale to larger problems?

Teyssier and Koller [2005] give a clever search and score

Submitted to the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022). **To be used for reviewing only**.

procedure, "Ordering Search", over variable permutations, pointing out that when two adjacent variables in a permutation are swapped, only local scores for the swapped variables need to be recalculated, the rest of the score calculation remains unchanged—this swapping operation is called an adjacency transposition. The Ordering Search algorithm greedily traverses the space of permutations with adjacency transpositions using a hill-climbing approach, random restarts, and a tabu list. However, they do not give any consistency guarantees.

The ESP ("Edge Sparsest Permutation") algorithm of Solus et al. [2021] iterates upon the Ordering Search algorithm by greedily traversing the space of permutations using weakly decreasing sequences of adjacent transpositions (found by depth first search (DFS)) to achieve asymptotic correctness. In addition, their TSP ("Triangle Sparsest Permutation") algorithm uses the theory of Chickering [2002] to navigate the space of DAGs, more efficiently, under stronger assumptions. Notably, SP, ESP, and TSP all assume the data generating distribution is a graphoid, which simplifies the computation of DAG construction considerably, since it is no longer NP-hard and can be done in polynomial time. A simulation study using a Python implementation of TSP [Solus et al., 2021] suggests that this procedure is fast, but has difficulty scaling accurately to moderate or large sized graphs [Lu et al., 2021].

To address the scaling problem for both accuracy and timing, in this paper we explore different ways of traversing the space of permutations that get closer to the performance and assumption relaxation of Raskutti and Uhler while maintaining scalability. As part of this effort, we also use the "Grow-Shrink" algorithm from Margaritis and Thrun [1999] to learn the DAG.

In what follows, we give an elaboration of the theoretical background of our set of permutation-based procedures, GRaSP ("Greedy Relaxations of Sparsest Permutation"). GRaSP has three tiers, GRaSP_0 (equivalent to TSP), GRaSP_1 (equivalent to ESP), and GRaSP_2 (a novel relaxation); we show how moving from a lower tier to a higher tier results in a gradual theoretical relaxation of the permutation search space and thus an improvement in accuracy. We then follow this with a study of oracle behavior for GRaSP_0, GRaSP_1, and GRaSP_2 on exhaustive lists of independence models with violations of faithfulness for all 4-variable regular Gaussian and positive discrete distributions and all 5-variable unfaithful DAGs with added marginal independencies between exogenous and endogenous variables. We also give a detailed simulation study for the linear, Gaussian case for larger possibly dense models of up to 100 variables, with consistently accurate results using GRaSP_2. We then give a conclusion and discussion where we point out areas of immediate future work.

2 BACKGROUND

Throughout this paper, italicized letters are used to denote variables (e.g., X_1, Y) and boldfaced letters for sets of variables (e.g., $\mathbf{X}$). Graphical definitions and notations related to directed acyclic graphs (DAGs) are provided in Appendix A.1. A DAG $\mathcal{G}$ over a set of measured variables $\mathbf{V} = \{X_1, ..., X_m\}$ consists of m vertices $\mathbf{v} = \{1, ..., m\}$ where each vertex i associates to the variable X_i, and each directed edge between two distinct vertices $j \to k$ represents the direct causal influence from X_j to X_k. We write $\mathbf{i} \perp_{\mathcal{G}} \mathbf{j} \,|\, \mathbf{k}$ to denote the *d-separation* relation between $\mathbf{i}$ and $\mathbf{j}$ given $\mathbf{k}$ in $\mathcal{G}$ for any pairwise disjoint subsets of vertices $\mathbf{i}, \mathbf{j}, \mathbf{k} \subseteq \mathbf{v}$. Similarly, given a joint probability distribution $\mathcal{P}$ over $\mathbf{V}$, denote $\mathbf{X} \perp\!\!\!\perp_{\mathcal{P}} \mathbf{Y} \,|\, \mathbf{Z}$ as the *conditional independence* (CI) relation between $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ for any pairwise disjoint subsets of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$.

A *model* is a pair $(\mathcal{G}, \mathcal{P})$ where $\mathcal{G}$ is a DAG and $\mathcal{P}$ is a joint probability distribution over the same set of measured variables $\mathbf{V}$. We use $(\mathcal{G}^*, \mathcal{P})$ to refer to the *true* data-generating model. Certain standard properties of a model can be defined in terms of the d-separation relations in $\mathcal{G}$ and the CI relations in $\mathcal{P}$. Denote $\mathtt{I}(\mathcal{G}) = \{\langle \mathbf{X_j}, \mathbf{X_k} \,|\, \mathbf{X_l} \rangle : \mathbf{j} \perp_{\mathcal{G}} \mathbf{k} \,|\, \mathbf{l}\}$ where $\mathbf{X_i} = \{X_j \in \mathbf{V} : j \in \mathbf{i}\}$ for every $\mathbf{i} \subseteq \mathbf{v}$, and $\mathtt{I}(\mathcal{P}) = \{\langle \mathbf{X}, \mathbf{Y} \,|\, \mathbf{Z} \rangle : \mathbf{X} \perp\!\!\!\perp_{\mathcal{P}} \mathbf{Y} \,|\, \mathbf{Z}\}$. Let $\mathtt{DAG}(\mathbf{V})$ be the set of all possible DAGs over $\mathbf{V}$.

Definition 2.1 *For any joint probability distribution $\mathcal{P}$ over $\mathbf{V}$, define $\mathtt{CMC}(\mathcal{P}) = \{\mathcal{G} \in \mathtt{DAG}(\mathbf{V}) : \mathtt{I}(\mathcal{G}) \subseteq \mathtt{I}(\mathcal{P})\}$ as the set of Markov DAGs. $(\mathcal{G}^*, \mathcal{P})$ satisfies the Markov assumption if $\mathcal{G}^* \in \mathtt{CMC}(\mathcal{P})$.*

Definition 2.2 *For any joint probability distribution $\mathcal{P}$, define $\mathtt{CFC}(\mathcal{P}) = \{\mathcal{G} \in \mathtt{CMC}(\mathcal{P}) : \mathtt{I}(\mathcal{P}) \subseteq \mathtt{I}(\mathcal{G})\}$ as the set of faithful DAGs. $(\mathcal{G}^*, \mathcal{P})$ satisfies the faithfulness assumption if $\mathcal{G}^* \in \mathtt{CFC}(\mathcal{P})$.*

Let $\mathtt{MEC}(\mathcal{G})$ be the *Markov equivalence class* (MEC) of $\mathcal{G}$ such that $\mathtt{I}(\mathcal{G}) = \mathtt{I}(\mathcal{G}')$ for each $\mathcal{G}' \in \mathtt{MEC}(\mathcal{G})$. One crucial goal of causal search is the identification of $\mathtt{MEC}(\mathcal{G}^*)$ from $\mathcal{P}$. A causal search algorithm is said to be *correct* if its output DAG (or the DAG induced by its output) is in $\mathtt{MEC}(\mathcal{G}^*)$. All known causal search algorithms that are correct assume the Markov assumption, and some classical algorithms (e.g., GES) assume faithfulness as well. Nevertheless, as pointed out by Uhler et al. [2013], learning CI relations from data by hypothesis testing is error-prone, and almost-violations of faithfulness are common. This motivates the exploration of causal search algorithms, which rely on assumptions strictly weaker than faithfulness. We call these assumptions *causal razors*.

One recent approach proposed by Raskutti and Uhler [2018] is the *SP* algorithm, which identifies the set of *sparsest permutations* defined over $\mathbf{v}$ under the following assumption. Let $\mathtt{E}(\mathcal{G})$ be the set of directed edges in a DAG $\mathcal{G}$.

Definition 2.3 *For any joint probability distribution $\mathcal{P}$, define* $\mathtt{Fr}(\mathcal{P}) = \{\mathcal{G} \in \mathtt{CMC}(\mathcal{P}) : \neg\exists\mathcal{G}' \in \mathtt{CMC}(\mathcal{P})$ *s.t.* $|\mathtt{E}(\mathcal{G}')| < |\mathtt{E}(\mathcal{G})|\}$ *and* $\mathtt{uFr}(\mathcal{P}) = \{\mathcal{G} \in \mathtt{Fr}(\mathcal{P}) : \neg\exists\mathcal{G}' \in \mathtt{Fr}(\mathcal{P})$ *s.t.* $\mathcal{G}' \notin \mathtt{MEC}(\mathcal{G})\}$ *as the sets of frugal DAGs and uniquely frugal, or u-frugal, DAGs respectively.* $(\mathcal{G}^*, \mathcal{P})$ *satisfies the u-frugality assumption if* $\mathcal{G}^* \in \mathtt{uFr}(\mathcal{P})$.[1]

SP is shown to be correct under the assumption of u-frugality. Below we introduce some necessary notations of permutation-based algorithms. To begin with, we refer the readers to Appendix **A.2** for the *graphoid properties*. Generally speaking, every joint probability distribution is a *semigraphoid*, strictly positive distributions are *graphoids*, and regular Gaussian distributions are *compositional graphoids*.

Given $\mathbf{V} = \{X_1, ..., X_m\}$, let $\Pi(\mathbf{v})$ be the set of all *permutations* over $\mathbf{v} = \{1, ..., m\}$. For each $\pi \in \Pi(\mathbf{v})$, let π_i be the i-th vertex in π, $\pi[i]$ be the index of vertex i in π (s.t. $\pi_{\pi[i]} = i$), and $\mathtt{Pre}(i, \pi) = \{\pi_j : 1 \leq j < \pi[i]\}$ be the set of vertices that precede i's index in π. We say that $\pi \in \Pi(\mathbf{v})$ is a *causal order* of $\mathcal{G} \in \mathtt{DAG}(\mathbf{V})$ if $j \in \mathtt{Pre}(i, \pi)$ for each $i \in \mathbf{v}$ and each $j \in \mathtt{An}(i, \mathcal{G})$ (i.e., the set of i's *ancestors* in $\mathcal{G}$). Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, each $\pi \in \Pi(\mathbf{v})$ induces a DAG $\mathcal{G}_\pi$ satisfying the following condition:

$$j \in \mathtt{Pre}(k, \pi) \text{ and } X_j \not\perp\!\!\!\perp_{\mathcal{P}} X_k \mid \mathbf{X}_{\mathtt{Pre}(k,\pi)\setminus\{j\}} \Leftrightarrow (j \to k) \in \mathtt{E}(\mathcal{G}_\pi). \tag{RU}$$

(RU) is the method of constructing a DAG from π and $\mathcal{P}$ discussed in [Raskutti and Uhler, 2018]. It is derived from a more general method in [Verma and Pearl, 1988] which only requires $\mathcal{P}$ to be semigraphoid. The two methods will be compared in Appendix **A.3**. But we refer $\mathcal{G}_\pi$ to the DAG induced from π and the graphoid $\mathcal{P}$ using (RU) unless specified otherwise. Obviously, π is a causal order of $\mathcal{G}_\pi$. Below shows an important feature of $\mathcal{G}_\pi$.

Definition 2.4 *For any joint probability distribution $\mathcal{P}$, define* $\mathtt{SGS}(\mathcal{P}) = \{\mathcal{G} \in \mathtt{CMC}(\mathcal{P}) : \neg\exists\mathcal{G}' \in \mathtt{CMC}(\mathcal{P})$ *s.t.* $\mathtt{E}(\mathcal{G}') \subset \mathtt{E}(\mathcal{G})\}$ *as the set of SGS-minimal DAGs.*[2]

Theorem 2.5 *[Appendix **A.3**] Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, $\mathcal{G}_\pi$ induced by π using (RU) is Markov and SGS-minimal for any $\pi \in \Pi(\mathbf{v})$.*

SP identifies the sparsest permutation $\hat{\pi} = \min_{\pi\in\Pi(\mathbf{v})} |\mathtt{E}(\mathcal{G}_\pi)|$, and thus $\mathcal{G}_{\hat{\pi}} \in \mathtt{MEC}(\mathcal{G}^*)$ when u-frugality is satisfied. Nevertheless, SP needs to examine all $|\mathbf{v}|!$ permutations in $\Pi(\mathbf{v})$ to identify the sparsest one and hence lacks scalability. Solus et al. [2021] introduce a greedy version of SP, namely *Triangle SP* (TSP), which is proven to be correct under faithfulness.[3] Below, we provide a quick and simple sketch of TSP's correctness under faithfulness.

TSP borrows the *Chickering algorithm* in [Chickering, 2002] to perform their *depth-first search* (DFS) procedure. For each vertex $i \in \mathbf{v}$, let $\mathtt{Pa}(i, \mathcal{G})$ be the set of *parents* in $\mathcal{G}$. A directed edge $i \to j$ is *covered* in $\mathcal{G}$ if $\mathtt{Pa}(i, \mathcal{G}) = \mathtt{Pa}(j, \mathcal{G}) \setminus \{i\}$.

Theorem 2.6 *[Chickering, 2002] Given a set of variables $\mathbf{V}$, for every pair of DAGs $\mathcal{G}, \mathcal{H} \in \mathtt{DAG}(\mathbf{V})$, if $\mathtt{I}(\mathcal{H}) \subseteq \mathtt{I}(\mathcal{G})$, there exists a sequence of DAGs, call it a Chickering sequence $\mathfrak{C} = \langle \mathcal{H} = \mathcal{G}^1, \mathcal{G}^2, ..., \mathcal{G}^k = \mathcal{G}\rangle$ (from $\mathcal{H}$ to $\mathcal{G}$) s.t. $\mathtt{I}(\mathcal{G}^i) \subseteq \mathtt{I}(\mathcal{G}^{i+1})$ and $\mathcal{G}^{i+1}$ is obtained from $\mathcal{G}^i$ by either reversing a covered edge or deleting a directed-edge for each $1 \leq i < k$.*[4]

Definition 2.7 *For any joint probability distribution $\mathcal{P}$, define* $\mathtt{Pm}(\mathcal{P}) = \{\mathcal{G} \in \mathtt{CMC}(\mathcal{P}) : \neg\exists\mathcal{G}' \in \mathtt{CMC}(\mathcal{P})$ *s.t.* $\mathtt{I}(\mathcal{G}) \subset \mathtt{I}(\mathcal{G}')\}$ *and* $\mathtt{uPm}(\mathcal{P}) = \{\mathcal{G} \in \mathtt{Pm}(\mathcal{P}) : \neg\exists\mathcal{G}' \in \mathtt{Pm}(\mathcal{P})$ *s.t.* $\mathcal{G}' \notin \mathtt{MEC}(\mathcal{G})\}$ *as the sets of P-minimal DAGs and unique P-minimal DAGs respectively.* $(\mathcal{G}^*, \mathcal{P})$ *satisfies the u-P-minimality assumption if* $\mathcal{G}^* \in \mathtt{uPm}(\mathcal{P})$.

Theorem 2.8 *For any joint probability distribution $\mathcal{P}$,*

(a) [Forster et al., 2020] $\mathtt{CFC}(\mathcal{P}) \subseteq \mathtt{Fr}(\mathcal{P}) \subseteq \mathtt{Pm}(\mathcal{P})$*;*

(b) [Zhang, 2013] $\mathtt{CFC}(\mathcal{P}) = \mathtt{Pm}(\mathcal{P}) = \mathtt{MEC}(\mathcal{G}^*)$ *if faithfulness holds.*

A sequence of DAGs $\langle \mathcal{G}^1, ..., \mathcal{G}^k\rangle$ is said to be *weakly decreasing* if $|\mathtt{E}(\mathcal{G}^i)| \geq |\mathtt{E}(\mathcal{G}^{i+1})|$ for each $1 \leq i < k$. Obviously, every Chickering sequence as defined above is a weakly decreasing sequence. Given an arbitrary initial permutation $\pi \in \Pi(\mathbf{v})$, TSP uses DFS to search for a Chickering sequence from $\mathcal{G}_\pi$ to some SGS-minimal DAG $\mathcal{G}_\tau$ where $|\mathtt{E}(\mathcal{G}_\pi)| > |\mathtt{E}(\mathcal{G}_\tau)|$, and update $\mathcal{G}_\pi$ as $\mathcal{G}_\tau$ until no such $\mathcal{G}_\tau$ is found. TSP guarantees that the returned $\hat{\mathcal{G}}_\pi$ is P-minimal. When faithfulness holds, $\hat{\mathcal{G}}_\pi \in \mathtt{Pm}(\mathcal{P}) = \mathtt{Fr}(\mathcal{P}) = \mathtt{CFC}(\mathcal{P}) = \mathtt{MEC}(\mathcal{G}^*)$, and hence GSP is correct.

Notice that the identification of a Chickering sequence from $\mathcal{G}_\pi$ to a P-minimal $\mathcal{G}_\tau$ is essentially a DAG-based operation

[1] This assumption is named as *sparsest Markov representation* (SMR) in [Raskutti and Uhler, 2018].

[2] SGS-minimality is also known as *minimal I-map* in the literature. But we follow Zhang [2013] to disambiguate different kinds of minimality conditions.

[3] In [Solus et al., 2021], *Greedy SP* (GSP) is an operational version of TSP which imposes a depth bound on the depth-first search procedure and a parameter specifying the number of runs on selecting an arbitrary initial permutation. They claimed that TSP can be correct even when faithfulness fails. We examine their claim more carefully in Section **3** and Appendix **C**.

[4] The original theorem in [Chickering, 2002] is expressed in terms of addition of directed edges. This modification helps by indicating that every Chickering sequence is a weakly decreasing sequence. In addition, one can easily observe that there does not exist any Chickering sequence from $\mathcal{G}$ to $\mathcal{H}$ if $\mathtt{I}(\mathcal{G}) \not\subseteq \mathtt{I}(\mathcal{H})$.

using the Chickering algorithm. In the next section, we introduce a permutation-based operation to converge to a P-minimal DAG more quickly than the Chickering algorithm, and propose a class of greedy permutation-based algorithms which employs weaker assumptions than TSP does.

3 METHODS

In this section, we introduce a class of permutation-based algorithms with a generic name *Greedy Relaxations of Sparsest Permutation* (GRaSP). Three tiers of relaxation will be studied: GRaSP_0 is our basic algorithm which is logically equivalent to TSP but more efficient, GRaSP_1 relaxes the search criterion of GRaSP_0 while GRaSP_2 further relaxes that of GRaSP_1. This hierarchy allows the identification of $\texttt{MEC}(\mathcal{G}^*)$ under progressively weaker assumptions. All proofs are left in Appendix **B**-**E**. First, we introduce our characteristic permutation-based operation *tuck*.

Definition 3.1 *Consider any graphoid $\mathcal{P}$ over $\mathbf{V}$, and any $j, k \in \mathbf{v}$ where $\pi[j] < \pi[k]$. Rewrite π as $\langle \boldsymbol{\delta}_1, j, \boldsymbol{\delta}_2, k, \boldsymbol{\delta}_3 \rangle$ where each $\boldsymbol{\delta}_i$ is a (possibly empty) sub-sequence of π.*[5] *Let $\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}^c$ be the sub-sequences $\langle i \in \boldsymbol{\delta}_2 : i \in \texttt{An}(k, \mathcal{G}_\pi) \rangle$ and $\langle i \in \boldsymbol{\delta}_2 : i \notin \texttt{An}(k, \mathcal{G}_\pi) \rangle$. Define*

$$tuck(\pi, j, k) = \begin{cases} \langle \boldsymbol{\delta}_1, \boldsymbol{\gamma}, k, j, \boldsymbol{\gamma}^c, \boldsymbol{\delta}_3 \rangle & \text{if } (j \to k) \in \texttt{E}(\mathcal{G}_\pi) \\ \pi & \text{otherwise.} \end{cases}$$

Example 3.2 *Consider $\pi = \langle 1, 2, 3, 4, 5 \rangle$ where $\texttt{Pa}(4, \mathcal{G}_\pi) = \{1, 3\}$. After rewriting π with respect to the pair $(1, 4)$, we have $\boldsymbol{\delta}_2 = \langle 2, 3 \rangle$ and $\boldsymbol{\gamma} = \langle 3 \rangle$. It follows from $(1 \to 4) \in \texttt{E}(\mathcal{G}_\pi)$ that $tuck(\pi, 1, 4) = \langle 3, 4, 1, 2, 5 \rangle$.*

Example 3.3 *Consider $\pi = \langle 1, 2, 3, 4, 5 \rangle$ where $(1 \to 4) \in \texttt{E}(\mathcal{G}_\pi)$ is a covered edge. We have $\boldsymbol{\gamma} = \varnothing$ since $\texttt{An}(4, \mathcal{G}_\pi) = \{1\}$. Hence, $tuck(\pi, 1, 4) = \langle 4, 1, 2, 3, 5 \rangle$.*

The last example illustrates that $tuck(\pi, j, k)$ simply moves k to the left of j in π when $j \to k$ is a covered edge in the induced $\mathcal{G}_\pi$. We specify this operation as *covered tuck* (*ct*).

Definition 3.4 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, for any $\pi, \tau \in \Pi(\mathbf{v})$, τ is said to be a ct-mutation of π if there exist $j, k \in \mathbf{v}$ s.t. $(j \to k) \in \texttt{E}(\mathcal{G}_\pi)$ is covered and $\tau = tuck(\pi, j, k)$. Also, $\langle \pi^1, ..., \pi^m \rangle$ is said to be a ct-sequence if π^{i+1} is a ct-mutation of π^i for each $1 \leq i < m$, and $(\mathcal{G}_{\pi^i}, \mathcal{G}_{\pi^l})$ are pairwise distinct for any $1 \leq i < l \leq m$.*

Lemma 3.5 *[Appendix **B**] Given a graphoid $\mathcal{P}$, every ct-sequence $\mathfrak{T} = \langle \pi, ..., \tau \rangle$ is weakly decreasing and entails the existence of a Chickering sequence from $\mathcal{G}_\pi$ to $\mathcal{G}_\tau$. Also, $|\mathfrak{T}| \leq |\mathfrak{C}|$ for any Chickering sequence $\mathfrak{C}$ from $\mathcal{G}_\pi$ to $\mathcal{G}_\tau$.*

[5] To be precise, $\boldsymbol{\delta}_1 = \langle \pi_i : 1 \leq i < \pi[j] \rangle$, $\boldsymbol{\delta}_2 = \langle \pi_i : \pi[j] < i < \pi[k] \rangle$, and $\boldsymbol{\delta}_3 = \langle \pi_i : \pi[k] < i \leq |\pi| \rangle$.

In the last section, we mentioned that TSP used the Chickering algorithm to perform a DAG-based DFS. The lemma above motivates a different DFS procedure. We employ our permutation-based operation *tuck* such that DFS is run on ct-sequences instead of Chickering sequences. We will also discuss when a ct-sequence is strictly shorter than its corresponding Chickering sequence in Appendix **B**. Hence, this suggests that the DFS of the former is more efficient than that of the latter.

Algorithm 1: DFS: $dfs(\mathcal{P}, \pi, d, d_{cur}, t)$

Input: (a) $\mathcal{P}$: a graphoid over $\mathbf{V}$; (b) $\pi \in \Pi(\mathbf{v})$; (c) d: depth bound; (d) d_{cur}: recorder of the recursive call; (e) t: type of directed edges
Output: $\tau \in \Pi(\mathbf{v})$ where $score(\tau) \geq score(\pi)$

```
foreach (j → k) ∈ E^t(G_π) do
    τ ← tuck(π, j, k)
    if score(τ) = score(π) and d_cur < d then
        τ ← dfs(P, τ, d, d_cur + 1)
    if score(τ) > score(π) then
        return τ
return π
```

Algorithm 2: GRaSP_t: $grasp(\mathcal{P}, \pi, d, t)$

Input: (a) $\mathcal{P}$: a graphoid over $\mathbf{V}$; (b) $\pi \in \Pi(\mathbf{v})$; (c) d: depth bound; (d) t: tier of GRaSP
Output: $\tau \in \Pi(\mathbf{v})$ where $score(\tau) \geq score(\pi)$

```
if t ≠ 0 then
    π = grasp(P, π, d, t − 1)
τ ← π
do
    π ← τ
    τ ← dfs(P, π, d, 1, t)
while score(τ) > score(π)
return τ
```

Algorithm 1 is our proposed DFS in terms of ct-sequences. First, we use *negative edge count* as the scoring function in our oracle version of the algorithm such that $score(\pi) = -|\texttt{E}(\mathcal{G}_\pi)|$ where $\mathcal{G}_\pi$ is induced from π and $\mathcal{P}$.[6] d bounds the search depth of DFS. We assume that $d = |\mathbf{V}|!$ for now and call the corresponding algorithm *unbounded*. In the next section, we will examine some small number d in light of finite samples. Also, we assume that no permutation can be revisited in the DFS procedure. Next, $\texttt{E}^t(\mathcal{G}_\pi)$ is the crucial function distinguishing our three tiers of GRaSP in **Algorithm 2**.

[6] Negative edge count is a consistent scoring function as shown in [Raskutti and Uhler, 2018].

Definition 3.6 *Given a DAG $\mathcal{G}$, a directed edge $(j \to k) \in \mathtt{E}(\mathcal{G})$ is said to be singular if there does not exist any unidirectional directed path from j to k in $\mathcal{G}$ except $j \to k$.*

$$\mathtt{E}^t(\mathcal{G}) = \begin{cases} \textit{covered edges in } \mathtt{E}(\mathcal{G}) & \textit{if } t = 0 \\ \textit{singular edges in } \mathtt{E}(\mathcal{G}) & \textit{if } t = 1 \\ \mathtt{E}(\mathcal{G}) & \textit{if } t = 2 \end{cases}$$

Theorem 3.7 *[Appendix B] Given a graphoid $\mathcal{P}$ over $\mathbf{V}$ and any $\pi \in \Pi(\mathbf{v})$, if $\mathcal{G}_\pi \notin \mathtt{Pm}(\mathcal{P})$, then there exists a ct-sequence $\mathfrak{T} = \langle \pi, ..., \tau \rangle$ s.t. $\mathtt{I}(\mathcal{G}_\pi) \subset \mathtt{I}(\mathcal{G}_\tau)$ and $\mathcal{G}_\tau \in \mathtt{Pm}(\mathcal{P})$.*

Readers can easily verify that $\mathtt{E}^0(\mathcal{G}) \subseteq \mathtt{E}^1(\mathcal{G}) \subseteq \mathtt{E}^2(\mathcal{G})$ holds for any DAG $\mathcal{G}$. Consider $\mathtt{E}^0(\mathcal{G})$ in particular. Given an arbitrary initial permutation π, **Algorithm 1** performs a greedy procedure to identify a ct-sequence from π. **Theorem 3.7** guarantees that $\hat{\tau}$ returned by **Algorithm 2** induces $\mathcal{G}_{\hat{\tau}} \in \mathtt{Pm}(\mathcal{P})$. Therefore, the correctness of unbounded GRaSP$_0$ under faithfulness follows immediately from **Theorem 2.8**. Given that $\mathtt{Pm}(\mathcal{P}) = \mathtt{Fr}(\mathcal{P})$ under faithfulness, the permutation returned by GRaSP$_1$ and GRaSP$_2$ cannot be sparser than that returned by GRaSP$_0$. Hence, the correctness of unbounded GRaSP$_1$ and unbounded GRaSP$_2$ under faithfulness immediately follows.

The sample version of GRaSP can be obtained by substituting the graphoid $\mathcal{P}$ with an *i.i.d.* observational dataset $\mathcal{D}$, and *score*(π) with the BIC score of $\mathcal{G}_\pi$ from $\mathcal{D}$ (defined in Appendix E). Pointwise consistency under faithfulness directly follows from the *local consistency* of BIC.[7]

Corollary 3.8 *Unbounded GRaSP$_0$, GRaSP$_1$, and GRaSP$_2$ are correct and pointwise consistent under faithfulness.*

On the other hand, the difference among the three tiers of GRaSP becomes explicit when faithfulness is violated.

Theorem 3.9 *[Appendix B] Given a graphoid $\mathcal{P}$, TSP and unbounded GRaSP$_0$ are logically equivalent.*

Theorem 3.10 *[Appendix B] For any joint probability distribution $\mathcal{P}$, $\mathtt{CFC}(\mathcal{P}) = \mathtt{uPm}(\mathcal{P})$.*

Theorem 3.11 *[Appendix B] Given a graphoid $\mathcal{P}$, faithfulness is necessary for the correctness of TSP and unbounded GRaSP$_0$.*[8]

[7] See [Haughton, 1988] and [Chickering, 2002] for the (local) consistency of BIC.

[8] The theorem is also true for GES which relies on the Chickering algorithm to identify a P-minimal DAG. When faithfulness fails, GES could return a P-minimal DAG $\mathcal{G} \notin \mathtt{MEC}(\mathcal{G}^*)$. This observation entails that GES, TSP, and unbounded GRaSP$_0$ are all logically equivalent.

Theorem 3.10 entails that there exist multiple P-minimal DAGs belonging to different MECs when faithfulness fails. So, even if $\hat{\tau}$ returned by TSP or unbounded GRaSP$_0$ induces a P-minimal DAG, it could be the case that $\mathcal{G}_{\hat{\tau}} \in \mathtt{Pm}(\mathcal{P}) \setminus \mathtt{Fr}(\mathcal{P})$ such that $\hat{\tau}$ is not one of the sparsest permutations. This observation prompts the usage of GRaSP with a higher tier. Extending $\mathtt{E}^0(\cdot)$ to $\mathtt{E}^1(\cdot)$ and $\mathtt{E}^2(\cdot)$ allows the traversal from $\hat{\tau}$ to some τ' where $|\mathtt{E}(\mathcal{G}_{\tau'})| < |\mathtt{E}(\mathcal{G}_{\hat{\tau}})|$. Thus, relaxing the search criterion defined by $\mathtt{E}^t(\cdot)$ licenses a higher tier of GRaSP to reach a sparser permutation under unfaithfulness. Examples of this sort will be studied in Appendix D.

The hierarchy of GRaSP implies that a higher tier of GRaSP requires assumptions increasingly weaker than faithfulness. In Appendix C, we show that unbounded GRaSP$_1$ requires the same assumption as another algorithm discussed in [Solus et al., 2021] named *Edge SP* (ESP). As a consequence, unbounded GRaSP$_2$ is a relaxation beyond the two assumptions discussed in their work. That said, we are aware of cases where unbounded GRaSP$_2$ is incorrect under u-frugality. Such a counterexample will be studied in Appendix D.

We conclude this section by discussing how to use the DAG-inducing method in Verma and Pearl [1988] based on BIC scores. This facilitates our simulations done in the next section. Given a semigraphoid $\mathcal{P}$ over $\mathbf{V}$, each $\pi \in \Pi(\mathbf{v})$ induces a DAG $\mathcal{G}_\pi$ satisfying the following condition:

$$X_j \in \mathbf{M} \Leftrightarrow (j \to k) \in \mathtt{E}(\mathcal{G}_\pi) \qquad \text{(VP)}$$

where $\mathbf{M}$ is a *Markov boundary* of X_k relative to $\mathbf{X}_{\mathtt{Pre}(k,\pi)}$ (defined in Appendix A.3). **Lemma A.4** highlights that the DAGs induced by (VP) and (RU) are equivalent when $\mathcal{P}$ is a graphoid. But (VP) is preferred since we can estimate the *unique* Markov boundary by the *Grow-Shrink* (GS) algorithm from [Margaritis and Thrun, 1999] using BIC scores and avoid hypothesis testing needed in (RU).

Theorem 3.12 *[Appendix E] Consider an observational dataset $\mathcal{D}$ with n i.i.d. observations from a (compositional) graphoid $\mathcal{P}$ over $\mathbf{V}$. In the large sample limit of n, for any $X \in \mathbf{V}$ and any $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X\}$, (Grow-)Shrink algorithm can identify the unique Markov boundary of X relative to $\mathbf{Z}$ using BIC scores.*

We leave the discussion of the GS algorithm and its related proof in Appendix E. In the next section, we are going to evaluate the performance of GRaSP through (VP) and GS in light of finite samples.

4 SIMULATIONS

In this section, we review empirical results of unfaithful u-frugal independence models with respect to DAGs and

algorithmic performance on Gaussian distributed data generated under a variety of situations. References to the code and instantiated model (IM) with replicability instructions will be included on a GitHub site for the project, which will be made available on publication. Included will be a running version of GRaSP (in Java) as well as covariance matrices for simulated data. (A Python translation is planned.)

4.1 U-FRUGAL FAITHFULNESS VIOLATIONS

In what follows, we consider sets of u-frugal independence models that violate the causal faithfulness assumption generated from: regular Gaussian distributions over four variables [Šimecek, 2006a], discrete distributions over four variables satisfying the intersection graphoid axiom and the Spohn condition [Šimecek, 2006b] (see Appendix **A.2**), and unfaithful DAGs (uDAGs) over five variables where a path cancellation is inducing a marginal independence between an exogenous variable and an endogenous variable (see Appendix **F**).[9]

We evaluate the capabilities of GRaSP_0, GRaSP_1, and GRaSP_2 to recover the u-frugal DAG from an independence oracle on these sets of independence models. We say that GRaSP_t recovers the u-frugal unfaithful model if it can do so from every initial permutation. Notice that if GRaSP_t can reach the u-frugal unfaithful model from every initial permutation, then the correctness of the algorithm will be independent of the DFS implementation.

	GRaSP_0	GRaSP_1	GRaSP_2	Total
Gaussian	0	7	10	10
Discrete	0	79	84	84
uDAGs	0	15	36	42

Table 1: Number of u-frugal unfaithful models recovered by GRaSP_0, GRaSP_1, and GRaSP_2. A model is only considered to be recovered if it is recovered from every initial permutation.

Table 1 tabulates examples where: GRaSP_0 fails but GRaSP_1 succeeds, and GRaSP_1 fails but GRaSP_2 succeeds; all of GRaSP_2's are contained within GRaSP_1's failures and all of GRaSP_1's failures are contained within GRaSP_0's failures. These results support the claims in **Corollary D.3**.

[9]The first two sets can be found at `http://5r.matfyz.cz/skola/models`. These sets were chosen over their more general counterparts because the more general sets contain independence models that are not graphoids. Furthermore, the discrete set contains the independence models for all positive discrete dsitributions.

4.2 LINEAR GAUSSIAN SIMULATIONS

We also studied the linear Gaussian case by varying simulations parameters around a configuration with 60 variables, an average degree of 6, and a sample size of 1,000. In Figure 2, we vary the number of measured variables from 20 to 100 with values 20, 30, 40, 50, 60, 70, 80, 90, and 100. In Figure 1, we vary the average degree from 2 to 10 with values 2, 3, 4, 5, 6, 7, 8, 9, and 10. For Figure 3, we vary the sample size from 200 to 100,000, with values 200, 500, 1,000, 2,000, 5,000, 10,000, 20,000, 50,000, and 100,000. In all cases, we draw coefficient values uniformly from $U(-1, 1)$ and incorporate independent additive exogenous noise distributions set to $N(0, 1)$. All statistics are averaged over 20 independent runs. Finally, in Figure 4, we give running times for our Java implementation of five algorithms, as follows: GRaSP_2, GRaSP_1, GRaSP_0, fGES, and PC. All of these but PC used BIC with a parameter penalty multiplier of 2 as a score; PC used partial correlation with a significance threshold of 0.001 as a conditional independence test. For GRaSP's parameters, we set depth to 3. Additionally, to maximize computational efficiency, we do not allow GRaSP to perform uncovered and non-singular tuck at any depth other than one[10]. In all cases, we follow the procedure set out in the text of running lower tiers of GRaSP before running higher tiers of GRaSP to guarantee consistent improvement of statistics.

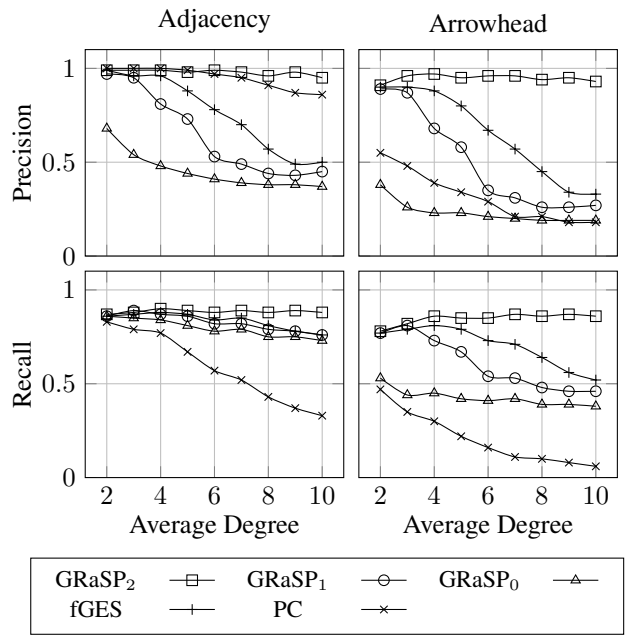


Figure 1: Average degree varied, measured variables fixed to 60, sample size fixed to 1,000.

[10]In the Java implementation of the algorithm, we include parameters for uncovered depth and non-singular depth to provide the user with more control over this heuristic.

In these figures, precision = $TP/(TP+FP)$ and recall = $TP/(TP+FN)$, where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. We give precision and recall statistics for adjacencies and arrowheads separately. For adjacencies, true (false) adjacencies are pairs of vertices that are (not) adjacent in the generative graphical model, and positive (negative) adjacencies are pairs of vertices that are (not) adjacent in the estimated graphical model for each algorithm, respectively. For arrowhead statistics, a true arrowhead is a directed edge in the CPDAG[11] of the generative graphical model and a positive arrowhead is a directed edge in the CPDAG of the estimated DAG, with negative and false arrowheads indicating the absence of these directed edges in their respective CPDAGs.

Figure 1 shows that algorithmic performance is strongly dependent on the average degree. While the compared algorithms generally perform well on sparse models, their performance drops off as the density increases. The exception is GRaSP_2, which dominates this group of algorithms, with a strong performance for both adjacencies and arrowheads as average degree is increased.

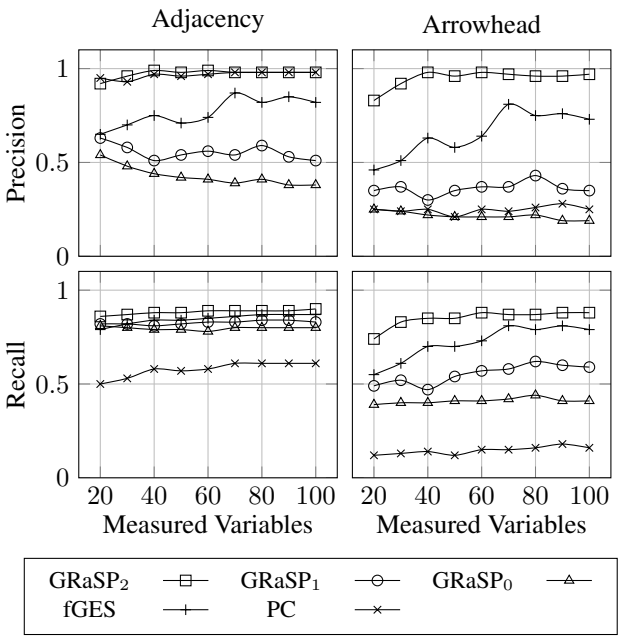


Figure 2: Measured variables varied, average degree fixed to 6, sample size fixed to 1,000.

Figure 2 shows the result of varying the number of measured variables. Notably, increasing the number of measured variables while holding the average degree constant decreases graph density. We see upward trends for some arrowhead statistics corresponding to this decrease in density. Again, GRaSP_2 dominates this group of algorithms, with strong precision and recall for both adjacencies and arrowheads.

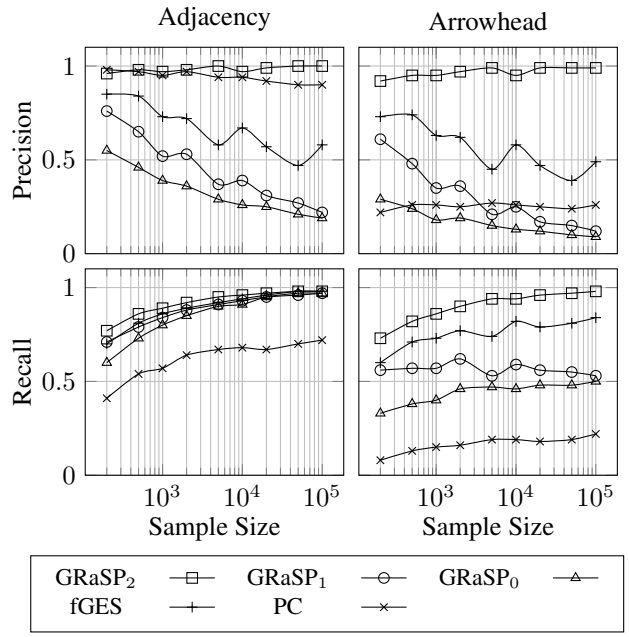


Figure 3: Sample size varied, measured variables fixed to 60, average degree fixed to 6.

All the algorithms compared in these simulations claim pointwise consistency, however, only GRaSP_2 seems to back up these claims in Figure 3. This might suggest that GRaSP_2 is better equipped to handle almost-violations of faithfulness in linear Gaussian models. As with previous figures, GRaSP_2 dominates this group of algorithms for precision and recall for both adjacencies and arrowheads for all sample sizes studied.

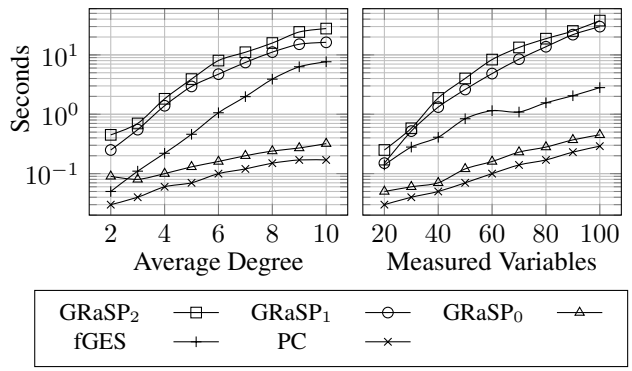


Figure 4: Measured variables fixed to 60 when not varied, averaged degree fixed to 6 when not varied, sample size fixed to 1,000.

Figure 4 shows that all the algorithms on average return in under a minute for the studied scenarios. However, given

[11] A CPDAG (a.k.a. "pattern") is a graphical representation of a Markov equivalence class for a DAG graphical equivalence class. See [Spirtes et al., 2000] for details.

the log scale, it should be noted that the computation time for $GRaSP_2$ increases exponentially with respect to the average degree of the graph and with respect to the number of measured variables. Other algorithms see similar slowdowns, but, other than $GRaSP_1$, none of the other algorithms experience as significant of a slow-down.[12]

In this paper, we focused on algorithms that can run on a 100 variable problem in a reasonable amount of time on a laptop. However, we would be remiss if we do not mention a recent algorithm by Lu et al. [2021] called Triplet A* that performs as well as, if not better, than $GRaSP_2$. We declined to directly compare the Triple A* algorithm in our Figures because it was unable to finish our simulations in reasonable time; for instance, the point Lu et al. give in their Figure 6 for the 60-variable, average degree 5 case was already as slow as could be managed (personal communication); we took our simulations out to an average degree of 10. In lieu of this, we include on our GitHub site results for our compared algorithms using Lu et al.'s simulation parameters for their Figures 3 and 6, for direct comparison.

5 DISCUSSION

Permutation-based reasoning in designing causal search algorithms is increasingly influential in the literature, including the methods from Teyssier and Koller [2005] and Raskutti and Uhler [2018]. We propose a class of algorithms under the generic name GRaSP characterized by an efficient permutation-based operation, *tuck*. All tiers of GRaSP are shown to be correct and pointwise consistent under the assumption of faithfulness. Also, we show that the two lower tiers of GRaSP are logically equivalent to the algorithms TSP and ESP discussed in [Solus et al., 2021]. We further prove that the final tier of GRaSP assumes a strictly weaker assumption than its lower-tier counterparts and demonstrate that it outperforms the lower-tier algorithms and two other state-of-the-art causal search algorithms, PC and fGES, in simulations.

Discussion of GRaSP can be extended in several directions. First, we have already begun to explore even higher tiers of GRaSP which relax the search criterion even further. Figure 1 suggests that GRaSP may provide tools helpful for the discussion of dense graph search. Given the hierarchy of GRaSP, higher tiers will hopefully improve the performance statistics and employ weaker assumption than the existing tiers. Ultimately, we hope to develop a tier of GRaSP that is correct under u-frugality alone.

[12] All simulations in this paper were run on a MacBook Pro laptop computer, M1, 2020, with 16G of RAM, using the Corretto 17 Java SDK. Memory is the main resource constraint on the procedure, which is needed for caching scores; for analyses significantly larger than the ones studied a supercomputer may be useful, not for multi-threading but for the larger available memory space.

Second, many advances have been made in the area of more or completely general modeling of data distributions, with corresponding improvements in accuracy of causal search for algorithms taking general modeling assumptions into account. It would be helpful to consider how such ideas can be incorporated into GRaSP. For example, Huang et al. [2018] show how a consistent general score can be incorporated into GES; it will be interesting to see whether GRaSP is able to show similar improvement in applicability when using such a score.

Third, we have analyzed Gaussian simulations in Section 4, but some simulation work needs to be done to show that GRaSP works well for discrete distributions (where the theory is already applicable) and also for mixed Gaussian/discrete distributions studied in [Andrews et al., 2019].

Fourth, the discussion of this paper is built upon the assumptions of causal sufficiency, that is, no latent common causes, and no selection bias. Causal search without these assumptions was pioneered by the FCI algorithm from Spirtes et al. [2000] and Zhang [2008]. To improve empirical performance of FCI, Ogarrio et al. [2016] initiated a hybrid algorithm GFCI which combines GES with FCI. To follow suit, we plan to explore an algorithm that incorporates GRaSP into GFCI (in place of GES), further improving this empirical performance.

Fifth, more direct comparisons to other algorithms need ideally to be done. As mentioned earlier, we include figures on our GitHub site using the simulation parameters in [Lu et al., 2021], Figures 3 and 6, so there is oblique comparison to the algorithms in those figures, including GES and PC in the PCALG package [Kalisch et al., 2012], Triplet A* [Lu et al., 2021], NOTEARS [Zheng et al., 2018], the GSP implementation in the Python causaldag package, LiNGAM [Shimizu et al., 2006], and MMHC [Tsamardinos et al., 2006]. The reader is invited to explore those comparisons.

Finally, while we have not taken up real data examples in this paper, it's useful to point out in a forward-looking way that improvements in the ability to handle latent and mixed continuous/discrete variables in a scalable and accurate causal search algorithm would put one in a good position to analyze a number of otherwise difficult real data examples. Accurate preliminary results consistent with ground truth using the suggested modification of GFCI for a number of mixed datasets from the Irvine Machine Learning Repository ([Dua and Graff, 2017]), for instance, suggest that this would be a good direction to look for new practical methods (cf. [Raghu et al., 2018]).

References

Bryan Andrews, Joseph Ramsey, and Gregory F Cooper. Learning high-dimensional directed acyclic graphs with mixed data-types. In *The 2019 ACM SIGKDD Workshop*

on Causal Discovery, pages 4–21. PMLR, 2019.

David Maxwell Chickering. A transformational characterization of equivalent Bayesian network structures. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, page 87–98, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3:507–554, 2002.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Malcolm Forster, Garvesh Raskutti, Reuben Stern, and Naftali Weinberger. The frugal inference of causal relations. *The British Journal for the Philosophy of Science*, 2020.

Dominique M. A. Haughton. On the choice of a model to fit data from an exponential family. *The Annals of Statistics*, 16:342 – 355, 1988.

Biwei Huang, Kun Zhang, Yizhu Lin, Bernhard Schölkopf, and Clark Glymour. Generalized score functions for causal discovery. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1551–1560, 2018.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of statistical software*, 47:1–26, 2012.

Robert E Kass and Paul W Vos. *Geometrical Foundations of Asymptotic Inference*. John Wiley & Sons, Hoboken, NJ, 2011.

Ni Y Lu, Kun Zhang, and Changhe Yuan. Improving causal discovery by optimal Bayesian network learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8741–8748, 2021.

Dimitris Margaritis and Sebastian Thrun. Bayesian network induction via local neighborhoods. *Advances in neural information processing systems*, 12, 1999.

Fatemeh Mohammadi, Caroline Uhler, Charles Wang, and Josephine Yu. Generalized permutohedra from probabilistic graphical models. *SIAM Journal on Discrete Mathematics*, 32:64–93, 2018.

Juan Miguel Ogarrio, Peter Spirtes, and Joe Ramsey. A hybrid causal search algorithm for latent variable models. In *Conference on probabilistic graphical models*, pages 368–379. PMLR, 2016.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

Vineet K Raghu, Joseph D Ramsey, Alison Morris, Dimitrios V Manatakis, Peter Sprites, Panos K Chrysanthis, Clark Glymour, and Panayiotis V Benos. Comparison of strategies for scalable causal discovery of latent variable models from mixed data. *International journal of data science and analytics*, 6:33–45, 2018.

Garvesh Raskutti and Caroline Uhler. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7, 2018.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7, 2006.

Petr Šimecek. Gaussian representation of independence models over four random variables. In *Proccedings of the International Conference on Computational Statistics*. Citeseer, 2006a.

Petr Šimecek. A short note on discrete representability of independence models. In *Proccedings of the Workshop on Probabilistic Graphical Models*, pages 287–292. Citeseer, 2006b.

Liam Solus, Yuhao Wang, and Caroline Uhler. Consistency guarantees for greedy permutation-based causal inference algorithms. *Biometrika*, 108:795–814, 2021.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Wolfgang Spohn. On the properties of conditional independence. In Paul Humphreys, editor, *Patrick Suppes: Scientific Philosopher: Volume 1. Probability and Probabilistic Causality*, pages 173–196. Springer Netherlands, 1994.

Milan Studený. *Probabilistic Conditional Independence Structures*. Information Science and Statistics. Springer, 2005.

Marc Teyssier and Daphne Koller. Ordering-based search: a simple and effective algorithm for learning Bayesian networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 584–590, 2005.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine learning*, 65:31–78, 2006.

Caroline Uhler, Garvesh Raskutti, Peter Bühlmann, and Bin Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, pages 436–463, 2013.

Thomas Verma and Judea Pearl. Causal networks: semantics and expressiveness. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 69–78, 1988.

Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172:1873–1896, 2008.

Jiji Zhang. A comparison of three Occam's razors for Markovian causal models. *The British journal for the philosophy of science*, 64:423–448, 2013.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. DAGs with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018.

A BACKGROUND MATERIALS

A.1 GRAPHICAL DEFINITIONS

A *directed graph* $\mathcal{G}$ over a set of measured variables $\mathbf{V} = \{X_1, ..., X_m\}$ consists of m vertices $\mathbf{v} = \{1, ..., m\}$ where each vertex $i \in \mathbf{v}$ associates to a variable $X_i \in \mathbf{V}$, and each edge in $\mathcal{G}$ is directed with the form $j \to k$ and no vertex has a directed edge to itself. A *directed path* $\mathfrak{p}$ is a sequence of vertices $\langle \mathbf{i}_1, \mathbf{i}_2, ..., \mathbf{i}_k \rangle$ for some $k \geq 2$ where $\mathbf{i}_j \in \mathbf{v}$ for each $1 \leq j \leq k$, and $\mathbf{i}_j$ and $\mathbf{i}_{j+1}$ are connected by a directed edge (i.e., $\mathbf{i}_j \to \mathbf{i}_{j+1}$ or $\mathbf{i}_{j+1} \to \mathbf{i}_j$). Such a path $\mathfrak{p}$ is *unidirectional* if $\mathbf{i}_j \to \mathbf{i}_{j+1}$ for each $1 \leq j < k$. A *directed acyclic graph* (DAG) is a directed graph where no vertex can have a unidirectional directed path to itself.

Denote $\mathtt{E}(\mathcal{G})$ as the set of directed edges in $\mathcal{G}$. Any pair of DAGs $\mathcal{G}_1, \mathcal{G}_2$ over the same set of variables $\mathbf{V}$ are equivalent if and only if $\mathtt{E}(\mathcal{G}_1) = \mathtt{E}(\mathcal{G}_2)$. Let $\mathtt{Pa}(j, \mathcal{G}) = \{k \in \mathbf{v} : (k \to j) \in \mathtt{E}(\mathcal{G})\}$ be the set of *parents* of j in $\mathcal{G}$, and $\mathtt{Ch}(j, \mathcal{G}) = \{k \in \mathbf{v} : (j \to k) \in \mathtt{E}(\mathcal{G})\}$ be the set of *children* of j in $\mathcal{G}$. $\mathtt{An}(j, \mathcal{G})$, the *ancestors* of j in $\mathcal{G}$, is defined by the transitive closure of $\mathtt{Pa}(j, \mathcal{G})$. Similarly, $\mathtt{De}(j, \mathcal{G})$, the *descendants* of j in $\mathcal{G}$, is defined by the transitive closure of $\mathtt{Ch}(j, \mathcal{G})$ and union with $\{j\}$ itself (i.e., j is its own descendant). Further let $\mathtt{Nd}(j, \mathcal{G}) = \mathbf{v} \setminus \mathtt{De}(j, \mathcal{G})$ be the set of j's *non-descendants*.

A pair of vertices $j, k \in \mathbf{v}$ are said to be *adjacent* in $\mathcal{G}$ if $(j \to k) \in \mathtt{E}(\mathcal{G})$ or $(k \to j) \in \mathtt{E}(\mathcal{G})$. For any triple of pairwise distinct vertices $i, j, k \in \mathbf{v}$, we say that (i, j, k) is *unshielded* if (i, j) and (j, k) are adjacent pairs in $\mathcal{G}$, but not (i, k). (i, j, k) forms a *triangle* if they are pairwise adjacent. If (i, j, k) is an unshielded triple or is a triangle, j is a *collider* (on the directed path $\langle i, j, k \rangle$) if $(i \to j), (k \to j) \in \mathtt{E}(\mathcal{G})$, and a *non-collider* otherwise. A directed path $\mathfrak{p}$ is a *trek* if it contains no collider.

For any $j, k \in \mathbf{v}$ and any $\mathbf{i} \subseteq \mathbf{v} \setminus \{j, k\}$, j and k are *d-connected* given $\mathbf{i}$ in $\mathcal{G}$ if there exists a directed path $\mathfrak{p}$ between j and k in $\mathcal{G}$ such that no non-collider on $\mathfrak{p}$ is in $\mathbf{i}$, and each collider l on $\mathfrak{p}$ or a l's descendant is in $\mathbf{i}$. j and k are *d-separated* given $\mathbf{i}$ in $\mathcal{G}$ if j and k are not d-connected given $\mathbf{i}$. For any disjoint subsets of vertices $\mathbf{j}, \mathbf{k}, \mathbf{i} \subseteq \mathbf{v}$, $\mathbf{j}$ and $\mathbf{k}$ are d-separated given $\mathbf{i}$ in $\mathcal{G}$ if j and k are d-separated by $\mathbf{i}$ in $\mathcal{G}$ for every $j \in \mathbf{j}$ and every $k \in \mathbf{k}$.

Given a model $(\mathcal{G}, \mathcal{P})$ over $\mathbf{V}$, $\mathcal{G}$ is said to be *local Markov* to $\mathcal{P}$ if $X_j \perp\!\!\!\perp_{\mathcal{P}} \mathbf{X}_{\mathtt{Nd}(j,\mathcal{G})} \setminus \mathbf{X}_{\mathtt{Pa}(j,\mathcal{G})} \,|\, \mathbf{X}_{\mathtt{Pa}(j,\mathcal{G})}$ for every $j \in \mathbf{v}$. It is a well-known fact that $\mathcal{G}$ is local Markov to $\mathcal{P}$ if and only if $\mathtt{I}(\mathcal{G}) \subseteq \mathtt{I}(\mathcal{P})$ (i.e., global Markov as defined by d-separation).

A.2 GRAPHOID PROPERTIES

For any pairwise disjoint sets of variables $\mathbf{W}, \mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$,

$$\begin{aligned}
\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z} &\Rightarrow \mathbf{Y} \perp\!\!\!\perp \mathbf{X} \,|\, \mathbf{Z} && (\textit{symmetry}) \\
\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \,|\, \mathbf{Z} &\Rightarrow (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \,|\, \mathbf{Z}) && (\textit{decomposition}) \\
\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \,|\, \mathbf{Z} &\Rightarrow \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z} \cup \mathbf{W} && (\textit{weak union}) \\
(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \,|\, \mathbf{Z} \cup \mathbf{Y}) &\Rightarrow \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \,|\, \mathbf{Z} && (\textit{contraction}) \\
(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z} \cup \mathbf{W}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \,|\, \mathbf{Z} \cup \mathbf{Y}) &\Rightarrow \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \,|\, \mathbf{Z} && (\textit{intersection}) \\
(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{Z}) \wedge (\mathbf{X} \perp\!\!\!\perp \mathbf{W} \,|\, \mathbf{Z}) &\Rightarrow \mathbf{X} \perp\!\!\!\perp \mathbf{Y} \cup \mathbf{W} \,|\, \mathbf{Z} && (\textit{composition})
\end{aligned}$$

A distribution $\mathcal{P}$ is a *semigraphoid* if $\mathtt{I}(\mathcal{P})$ is closed under *symmetry*, *decomposition*, *weak union*, and *contraction*. A semigraphoid $\mathcal{P}$ is a *graphoid* if $\mathtt{I}(\mathcal{P})$ is closed under *intersection*. A graphoid $\mathcal{P}$ is *compositional* if $\mathtt{I}(\mathcal{P})$ is closed under *composition*. See Chapter 2 of [Studený, 2005] for a more comprehensive study of graphoid properties. In addition, applications of *symmetry* in our upcoming proofs will be done implicitly for the sake of simplicity.

Additionally, Spohn [1994] notes that the following property necessarily holds in the independence models induced by positive discrete probability distributions. For any pairwise disjoint sets of variables $\mathbf{W}, \mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$,

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \,|\, \mathbf{W} \cup \mathbf{Z}) \wedge (\mathbf{W} \perp\!\!\!\perp \mathbf{Z} \,|\, \mathbf{X} \cup \mathbf{Y}) \wedge (\mathbf{W} \perp\!\!\!\perp \mathbf{Z} \,|\, \mathbf{X}) \;\Rightarrow\; [(\mathbf{W} \perp\!\!\!\perp \mathbf{Z} \,|\, \mathbf{Y}) \Leftrightarrow (\mathbf{W} \perp\!\!\!\perp \mathbf{Z} \,|\, \varnothing)] \qquad (\textit{Spohn condition})$$

A.3 DAG INDUCED FROM A PERMUTATION

Definition A.1 *Given a semigraphoid $\mathcal{P}$ over $\mathbf{V}$, for every $X \in \mathbf{V}$, we say that $\mathbf{M} \subseteq \mathbf{V}$ is a Markov blanket of X relative to $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X\}$ if*

(i) $\mathbf{M} \subseteq \mathbf{Z}$*;*

(ii) $X \perp\!\!\!\perp_{\mathcal{P}} (\mathbf{Z} \setminus \mathbf{M}) \,|\, \mathbf{M}$.

Such a Markov blanket $\mathbf{M}$ *is said to be a Markov boundary if it further satisfies the following condition:*

(iii) there does not exist $\mathbf{M}' \subset \mathbf{M}$ *s.t.* $X \perp\!\!\!\perp_{\mathcal{P}} (\mathbf{Z} \setminus \mathbf{M}') \,|\, \mathbf{M}'$.

Lemma A.2 *[Verma and Pearl, 1988] Given a graphoid* $\mathcal{P}$ *over* $\mathbf{V}$*, for every* $X \in \mathbf{V}$ *and every* $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X\}$*, there is a unique Markov boundary of* X *relative to* $\mathbf{Z}$.

In the following, we use $\mathtt{MB}_{\mathcal{P}}(X, \mathbf{Z})$ to refer to the unique Markov boundary of X relative to $\mathbf{Z}$. The subscript $\mathcal{P}$ will be suppressed if the underlying graphoid is clear from context.

Lemma A.3 *Given a graphoid* $\mathcal{P}$ *over* $\mathbf{V}$*, for every* $X \in \mathbf{V}$ *and every* $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X\}$*, if* $\mathbf{M}$ *is a Markov blanket of* X *relative to* $\mathbf{Z}$*, then* $\mathtt{MB}(X, \mathbf{Z}) \subseteq \mathbf{M}$.

Proof. Immediate from **Definition A.1** and **Lemma A.2**. □

Next, we restate the two methods of inducing a DAG from a permutation. Given a semigraphoid $\mathcal{P}$ over $\mathbf{V}$, each $\pi \in \Pi(\mathbf{v})$ induces a DAG satisfying the following condition:

$$X_j \in \mathbf{M} \Leftrightarrow (j \to k) \in \mathtt{E}(\mathcal{G}_\pi) \tag{VP}$$

where $\mathbf{M}$ is a Markov boundary of X_k relative to $\mathbf{X}_{\mathtt{Pre}(k,\pi)}$. (VP) is the construction of a *boundary DAG* in [Verma and Pearl, 1988]. On the other hand, given a graphoid $\mathcal{P}$ over $\mathbf{V}$, each $\pi \in \Pi(\mathbf{v})$ induces a DAG satisfying the following condition:

$$j \in \mathtt{Pre}(k, \pi) \text{ and } X_j \not\perp\!\!\!\perp_{\mathcal{P}} X_k \,|\, \mathbf{X}_{\mathtt{Pre}(k,\pi)\setminus\{j\}} \Leftrightarrow (j \to k) \in \mathtt{E}(\mathcal{G}_\pi). \tag{RU}$$

We want to show that the two DAG-inducing methods are equivalent when the underlying distribution is a graphoid.

Lemma A.4 *Given a graphoid* $\mathcal{P}$ *over* $\mathbf{V}$*, consider any* $\pi \in \Pi(\mathbf{v})$*. Let* $\mathcal{G}_\pi$ *be the DAG induced from* π *by (VP), and* $\mathcal{H}_\pi$ *be the DAG induced from* π *by (RU). Then* $\mathcal{G}_\pi = \mathcal{H}_\pi$.

Proof. We divide the proof into two directions: (VP) ⇒ (RU), and (VP) ⇐ (RU). Consider any $j, k \in \mathbf{v}$ where $\pi[j] < \pi[k]$ such that $j \in \mathtt{Pre}(k, \pi)$. Let $\mathbf{M}$ be the unique Markov boundary $\mathtt{MB}(X_k, \mathbf{X}_{\mathtt{Pre}(k,\pi)})$.

[⇒] Suppose that $(j \to k) \notin \mathtt{E}(\mathcal{G}_\pi)$. We have $X_j \notin \mathbf{M}$. By **Definition A.1** (ii), we then have,

$$X_k \perp\!\!\!\perp_{\mathcal{P}} ((\mathbf{X}_{\mathtt{Pre}(k,\pi)} \setminus \mathbf{M}) \setminus \{X_j\}) \cup \{X_j\} \,|\, \mathbf{M} \qquad \because X_k \perp\!\!\!\perp_{\mathcal{P}} \mathbf{X}_{\mathtt{Pre}(k,\pi)} \setminus \mathbf{M} \,|\, \mathbf{M} \tag{1}$$

$$X_k \perp\!\!\!\perp_{\mathcal{P}} X_j \,|\, \mathbf{M} \cup ((\mathbf{X}_{\mathtt{Pre}(k,\pi)} \setminus \mathbf{M}) \setminus \{X_j\}) \qquad \because (1), \text{weak union} \tag{2}$$

$$X_k \perp\!\!\!\perp_{\mathcal{P}} X_j \,|\, \mathbf{X}_{\mathtt{Pre}(k,\pi)\setminus\{j\}} \qquad \because (2) \tag{3}$$

where the last formula amounts to $(j \to k) \notin \mathtt{E}(\mathcal{H}_\pi)$ by (RU).

[⇐] Suppose that $(j \to k) \notin \mathtt{E}(\mathcal{H}_\pi)$. We have $X_k \perp\!\!\!\perp_{\mathcal{P}} X_j \,|\, \mathbf{X}_{\mathtt{Pre}(k,\pi)\setminus\{j\}}$. Let $\mathbf{M}'$ be $\mathbf{X}_{\mathtt{Pre}(k,\pi)\setminus\{j\}}$. We have $X_k \perp\!\!\!\perp_{\mathcal{P}} (\mathbf{X}_{\mathtt{Pre}(k,\pi)} \setminus \mathbf{M}') \,|\, \mathbf{M}'$ such that $\mathbf{M}'$ is a Markov blanket of X_k relative to $\mathbf{X}_{\mathtt{Pre}(k,\pi)}$. By **Lemma A.3**, $X_j \notin \mathbf{M} \subseteq \mathbf{M}'$ and therefore $(j \to k) \notin \mathtt{E}(\mathcal{G}_\pi)$ by (VP). □

Theorem A.5 *[Pearl, 1988] Given a semigraphoid* $\mathcal{P}$ *over* $\mathbf{V}$*,* $\mathcal{G}_\pi$ *induced by* π *using (VP) is Markov and SGS-minimal for any* $\pi \in \Pi(\mathbf{v})$.

Theorem 2.5 *Given a graphoid* $\mathcal{P}$ *over* $\mathbf{V}$*,* $\mathcal{G}_\pi$ *induced by* π *using (RU) is Markov and SGS-minimal for any* $\pi \in \Pi(\mathbf{v})$.

Proof. Immediate from **Lemma A.4** and **Theorem A.5**.[13] □

[13] The two DAG-inducing methods were not differentiated in [Raskutti and Uhler, 2018]. Thus, we provide a proof of **Theorem 2.5**.

B CORRECTNESS RESULTS

First, we introduce some permutation-based notations to facilitate our coming proofs. In this section, we refer $\mathcal{G}_\pi$ as the DAG induced by π from a graphoid $\mathcal{P}$ using (RU) unless specified otherwise.

Given a set of variables $\mathbf{V}$, consider any $\pi \in \Pi(\mathbf{v})$ and any pair $j, k \in \mathbf{v}$ where $\pi[j] < \pi[k]$. π can be written as $\langle \boldsymbol{\delta}_{<j}, j, \boldsymbol{\delta}_{j\sim k}, k, \boldsymbol{\delta}_{>k} \rangle$ such that $\boldsymbol{\delta}_{<j} = \langle \pi_i : 1 \leq i < \pi[j] \rangle$, $\boldsymbol{\delta}_{j\sim k} = \langle \pi_i : \pi[j] < i < \pi[k] \rangle$, and $\boldsymbol{\delta}_{>k} = \langle \pi_i : \pi[k] < i \leq |\pi| \rangle$. When $\boldsymbol{\delta}_{j\sim k} = \varnothing$, we say that j and k are π-*adjacent*. In that case, π can be written as $\langle \boldsymbol{\delta}_{<j}, j, k, \boldsymbol{\delta}_{>k} \rangle$ instead.

Definition B.1 *Given a set of variables $\mathbf{V}$, for any $\pi, \tau \in \Pi(\mathbf{v})$,*

(a) τ is said to be (j, k)-different from π for some $j, k \in \mathbf{v}$ if j and k are π-adjacent (i.e., $\pi = \langle \boldsymbol{\delta}_{<j}, j, k, \boldsymbol{\delta}_{>k} \rangle$) and $\tau = \langle \boldsymbol{\delta}_{<j}, k, j, \boldsymbol{\delta}_{>k} \rangle$;

(b) π and τ are said to be in adjacent transposition (AT) if they are (j, k)-different for some $j, k \in \mathbf{v}$.

Lemma B.2 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any $\mathcal{H} \in \mathtt{CMC}(\mathcal{P})$. If $\pi \in \Pi(\mathbf{v})$ is a causal order of $\mathcal{G}$, then $\mathtt{E}(\mathcal{G}_\pi) \subseteq \mathtt{E}(\mathcal{H})$. Also, $\mathcal{G}_\pi = \mathcal{H}$ if $\mathcal{H} \in \mathtt{SGS}(\mathcal{P})$.*

Proof. Consider any $k \in \mathbf{v}$ and $\mathtt{Nd}(k, \mathcal{H})$ (i.e., the set of k's non-descendants in $\mathcal{H}$). Since $\mathcal{H} \in \mathtt{CMC}(\mathcal{P})$, it follows that $X_k \perp\!\!\!\perp_{\mathcal{P}} \mathbf{X}_{\mathtt{Nd}(k,\mathcal{H})} \setminus \mathbf{X}_{\mathtt{Pa}(k,\mathcal{H})} \mid \mathbf{X}_{\mathtt{Pa}(k,\mathcal{H})}$. Also, we have $\mathtt{Pa}(k, \mathcal{H}) \subseteq \mathtt{Pre}(k, \pi) \subseteq \mathtt{Nd}(k, \mathcal{H})$ from π's being a causal order of $\mathcal{H}$. By *decomposition*, we have $X_k \perp\!\!\!\perp_{\mathcal{P}} \mathbf{X}_{\mathtt{Pre}(k,\pi)} \setminus \mathbf{X}_{\mathtt{Pa}(k,\mathcal{H})} \mid \mathbf{X}_{\mathtt{Pa}(k,\mathcal{H})}$ such that $\mathbf{X}_{\mathtt{Pa}(k,\mathcal{H})}$ is a Markov blanket of X_k relative to $\mathbf{X}_{\mathtt{Pre}(k,\pi)}$. By **Lemma A.3**, we have $\mathtt{MB}(X_k, \mathbf{X}_{\mathtt{Pre}(k,\pi)}) \subseteq \mathtt{Pa}(k, \mathcal{H})$. Consider $\mathcal{G}_\pi$ induced by (VP). The above entails that $\mathtt{E}(\mathcal{G}_\pi) \subseteq \mathtt{E}(\mathcal{H})$ since $\mathtt{Pa}(k, \mathcal{G}_\pi) = \mathtt{MB}(X_k, \mathbf{X}_{\mathtt{Pre}(k,\pi)}) \subseteq \mathtt{Pa}(k, \mathcal{H})$ for each $k \in \mathbf{v}$. Due to **Lemma A.4**, $\mathtt{E}(\mathcal{G}_\pi) \subseteq \mathtt{E}(\mathcal{H})$ still holds even if $\mathcal{G}_\pi$ is induced by (RU). Lastly, $\mathcal{G}_\pi = \mathcal{H}$ follows from **Definition 2.4** if $\mathcal{H} \in \mathtt{SGS}(\mathcal{P})$. □

Lemma B.3 *[Solus et al., 2021] Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any $\pi, \tau \in \Pi(\mathbf{v})$ where τ is (j, k)-different from π for some $j, k \in \mathbf{v}$. Then $\mathcal{G}_\pi = \mathcal{G}_\tau$ if and only if $X_j \perp\!\!\!\perp_{\mathcal{P}} X_k \mid \mathbf{X}_{\mathtt{Pre}(j,\pi)}$.*

Proof. Suppose that $X_j \not\perp\!\!\!\perp_{\mathcal{P}} X_k \mid \mathbf{X}_{\mathtt{Pre}(j,\pi)}$. By (RU), we have $(j \to k) \in \mathtt{E}(\mathcal{G}_\pi)$. Note that $(j \to k) \notin \mathtt{E}(\mathcal{G}_\tau)$ since $\tau[k] < \tau[j]$ and τ is a causal order of $\mathcal{G}_\tau$ by construction. Hence $\mathcal{G}_\pi \neq \mathcal{G}_\tau$.

On the other hand, suppose that $X_j \perp\!\!\!\perp_{\mathcal{P}} X_k \mid \mathbf{X}_{\mathtt{Pre}(j,\pi)}$. Since τ is (j, k)-different from π, we have $\pi = \langle \boldsymbol{\delta}_{<j}, j, k, \boldsymbol{\delta}_{>k} \rangle$ and $\tau = \langle \boldsymbol{\delta}_{<j}, k, j, \boldsymbol{\delta}_{>k} \rangle$ according to **Definition B.1** (a). By (RU), we know that $(k \to j) \notin \mathtt{E}(\mathcal{G}_\tau)$. Hence, π is a causal order of $\mathcal{G}_\tau$. By **Theorem 2.5**, $\mathcal{G}_\tau \in \mathtt{SGS}(\mathcal{P})$. Therefore, it follows from **Lemma B.2** that $\mathcal{G}_\tau = \mathcal{G}_\pi$. □

Lemma B.4 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any $\pi \in \Pi(\mathbf{v})$. Suppose that $\mathcal{G}_\pi$ contains a covered edge $j \to k$ where $\pi = \langle \boldsymbol{\delta}_{<j}, j, \boldsymbol{\delta}_{j\sim k}, k, \boldsymbol{\delta}_{>k} \rangle$. If $\tau = \langle \boldsymbol{\delta}_{<j}, j, k, \boldsymbol{\delta}_{j\sim k}, \boldsymbol{\delta}_{>k} \rangle$, then $\mathcal{G}_\pi = \mathcal{G}_\tau$.*

Proof. Since $j \to k$ is a covered edge in $\mathcal{G}_\pi$, it follows that $(i \to k) \notin \mathtt{E}(\mathcal{G}_\pi)$ for each $i \in \boldsymbol{\delta}_{j\sim k}$. Hence, $\mathcal{G}_\pi = \mathcal{G}_\tau$ can be obtained after $|\boldsymbol{\delta}_{j\sim k}|$ applications of **Lemma B.3**. □

Theorem B.5 *[Zhang, 2013] Given a set of variables $\mathbf{V}$, for any $\mathcal{G}, \mathcal{H} \in \mathtt{DAG}(\mathbf{V})$, if $\mathtt{E}(\mathcal{G}) \subseteq \mathtt{E}(\mathcal{H})$, then $\mathtt{I}(\mathcal{H}) \subseteq \mathtt{I}(\mathcal{G})$.*

Lemma B.6 *[Chickering, 1995] Consider any DAG $\mathcal{G}$. Let $\mathcal{G}'$ be the result of reversing $(i \to j) \in \mathtt{E}(\mathcal{G})$. Then $\mathcal{G}' \in \mathtt{MEC}(\mathcal{G})$ if and only if $i \to j$ is a covered edge.*

Lemma B.7 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any $\pi \in \Pi(\mathbf{v})$. Suppose that $(j \to k) \in \mathtt{E}(\mathcal{G}_\pi)$ is a covered edge, and let $\mathcal{H}$ be the DAG resulted from reversing $(j \to k)$ in $\mathcal{G}_\pi$. If $\tau = tuck(\pi, j, k)$, then*

(a) τ is a causal order of $\mathcal{H}$;

(b) $\mathtt{E}(\mathcal{G}_\tau) \subseteq \mathtt{E}(\mathcal{H})$;

(c) $|\mathtt{E}(\mathcal{G}_\tau)| \leq |\mathtt{E}(\mathcal{G}_\pi)|$;

(d) $\mathtt{I}(\mathcal{G}_\pi) \subseteq \mathtt{I}(\mathcal{G}_\tau)$.

Proof. A similar lemma has been shown in [Solus et al., 2021]. First, we write $\pi = \langle \boldsymbol{\delta}_{<j}, j, \boldsymbol{\delta}_{j\sim k}, k, \boldsymbol{\delta}_{>k} \rangle$ as usual. Consider $\pi' = \langle \boldsymbol{\delta}_{<j}, j, k, \boldsymbol{\delta}_{j\sim k}, \boldsymbol{\delta}_{>k} \rangle$. By **Lemma B.4**, we have $\mathcal{G}_\pi = \mathcal{G}_{\pi'}$. Note that $\tau = \textit{tuck}(\pi, j, k) = \langle \boldsymbol{\delta}_{<j}, k, j, \boldsymbol{\delta}_{j\sim k}, \boldsymbol{\delta}_{>k} \rangle$ because $\textit{tuck}(\pi, j, k)$ is a covered tuck. Thus, τ is (j, k)-different from π'. Also, since π' is a causal order of $\mathcal{G}_\pi$, it follows that τ is a causal order of $\mathcal{H}$ and thus (a) is proven.

Next, observe that $\mathtt{I}(\mathcal{G}_\pi) = \mathtt{I}(\mathcal{H})$ from **Lemma B.6**. From $\mathcal{G}_\pi \in \mathtt{CMC}(\mathcal{P})$ by **Theorem 2.5**, we know that $\mathcal{H} \in \mathtt{CMC}(\mathcal{P})$. Thus, (b) immediately follows from (a) and **Lemma B.2**. Also, (c) is entailed by $|\mathtt{E}(\mathcal{G}_\tau)| \leq |\mathtt{E}(\mathcal{H})| = |\mathtt{E}(\mathcal{G}_\tau)|$. Finally, by **Theorem B.5**, we have $\mathtt{I}(\mathcal{G}_\pi) = \mathtt{I}(\mathcal{H}) \subseteq \mathtt{I}(\mathcal{G}_\tau)$ as desired in (d). □

Lemma B.8 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any $\pi \in \Pi(\mathbf{v})$. If $\mathcal{G}_\pi \notin \mathtt{Pm}(\mathcal{P})$, there exists a ct-sequence from π to τ s.t. $\mathtt{I}(\mathcal{G}_\pi) \subset \mathtt{I}(\mathcal{G}_\tau)$.*

Proof. Suppose that $\mathcal{G}_\pi \notin \mathtt{Pm}(\mathcal{P})$. By **Definition 2.7**, there exists $\mathcal{H} \in \mathtt{CMC}(\mathcal{P})$ such that $\mathtt{I}(\mathcal{G}_\pi) \subset \mathtt{I}(\mathcal{H})$. By **Theorem 2.6**, there exists a Chickering sequence $\mathfrak{C} = \langle \mathcal{G}_\pi = \mathcal{G}^1, \mathcal{G}^2, ..., \mathcal{G}^m = \mathcal{H} \rangle$ such that $\mathtt{I}(\mathcal{G}^i) \subseteq \mathtt{I}(\mathcal{G}^{i+1})$ for $1 \leq i < m$. Without loss of generality, suppose that $\mathfrak{C}$ is the shortest Chickering sequence such that $\mathtt{I}(\mathcal{G}^1) = \mathtt{I}(\mathcal{G}^2) = ... = \mathtt{I}(\mathcal{G}^{m-1}) \subset \mathtt{I}(\mathcal{G}^m) \subseteq \mathtt{I}(\mathcal{P})$, also $\mathcal{G}^i$ and $\mathcal{G}^\ell$ are pairwise distinct for $1 \leq i < \ell \leq m$. We prove the lemma constructively from $\mathfrak{C}$.

---

1 $\pi^1 \leftarrow \pi$
2 $\mathfrak{T} \leftarrow \langle \pi^1 \rangle$
3 $i \leftarrow 1$
4 **while** $i < m - 1$ **do**
5 $\quad j, k \leftarrow$ the covered edge $(j \to k) \in \mathtt{E}(\mathcal{G}^i)$ s.t. $(k \to j) \in \mathtt{E}(\mathcal{G}^{i+1})$
6 $\quad \pi^{i+1} \leftarrow \textit{tuck}(\pi^i, j, k)$
7 $\quad$ Insert π^{i+1} to the end of $\mathfrak{T}$
8 $\quad$ **if** $\mathtt{I}(\mathcal{G}_{\pi^i}) \subset \mathtt{I}(\mathcal{G}_{\pi^{i+1}})$ **then**
9 $\quad\quad$ return $\mathfrak{T}$
10 $\quad i \leftarrow i + 1;$

We want to show that the construction of $\mathfrak{T} = \langle \pi^1, ..., \pi^k \rangle$ returned by the procedure above is our desired ct-sequence. First, by **Theorem 2.5**, $\mathcal{G}_{\pi^1} = \mathcal{G}_\pi = \mathcal{G}^1 \in \mathtt{SGS}(\mathcal{P})$. Next, we show that $m > 2$. By reductio, if $m = 2$, we have $\mathtt{I}(\mathcal{G}^1) \subset \mathtt{I}(\mathcal{G}^2)$ where $\mathcal{G}^2$ is obtained from deleting an edge in $\mathcal{G}^1$ such that $\mathtt{E}(\mathcal{G}^2) \subset \mathtt{E}(\mathcal{G}^1)$. However, this contradicts with $\mathcal{G}^1 \in \mathtt{SGS}(\mathcal{P})$ and $\mathcal{G}^m \in \mathtt{CMC}(\mathcal{P})$. Hence, $m > 2$.

Now we consider each call of the while-loop. Take the first call $i = 1$ such that $\mathtt{I}(\mathcal{G}^1) = \mathtt{I}(\mathcal{G}^2)$. By **Theorem 2.6**, $\mathcal{G}^1$ and $\mathcal{G}^2$ differ only by a covered edge $(j \to k) \in \mathtt{E}(\mathcal{G}^1)$. We then have $\pi^2 = \textit{tuck}(\pi^1, j, k)$ in line 6. It follows from **Lemma B.7** (d) that $\mathtt{I}(\mathcal{G}_{\pi^1}) = \mathtt{I}(\mathcal{G}^1) = \mathtt{I}(\mathcal{G}^2) \subseteq \mathtt{I}(\mathcal{G}_{\pi^2})$. Observe that $\mathfrak{T} = \langle \pi^1, \pi^2 \rangle$. Either $\mathtt{I}(\mathcal{G}_{\pi^1}) = \mathtt{I}(\mathcal{G}_{\pi^2})$ or $\mathtt{I}(\mathcal{G}_{\pi^1}) \subset \mathtt{I}(\mathcal{G}_{\pi^2})$. The desired ct-sequence is found if the latter holds because π^2 is a ct-mutation of π^1 (and hence $|\mathfrak{T}| < |\mathfrak{C}|$). Consider the case $\mathtt{I}(\mathcal{G}_{\pi^1}) = \mathtt{I}(\mathcal{G}_{\pi^2})$. By **Lemma B.7** (b), we have $\mathtt{E}(\mathcal{G}_{\pi^2}) \subseteq \mathtt{E}(\mathcal{G}^2)$. We know that $\mathtt{E}(\mathcal{G}_{\pi^2}) = \mathtt{E}(\mathcal{G}^2)$ because $\mathtt{I}(\mathcal{G}_{\pi^2}) = \mathtt{I}(\mathcal{G}^2)$ will be violated if $\mathtt{E}(\mathcal{G}_{\pi^2}) \subset \mathtt{E}(\mathcal{G}^2)$. Hence, if $\mathtt{I}(\mathcal{G}_{\pi^2}) = \mathtt{I}(\mathcal{G}^2)$ then $\mathcal{G}_{\pi^2} = \mathcal{G}^2 \in \mathtt{SGS}(\mathcal{P})$ due to **Theorem 2.5**. Obviously, $\mathcal{G}_{\pi^1} \neq \mathcal{G}_{\pi^2}$ due to the covered edge reversal. Next, we increment i by 1.

Notice that after running the procedure above, either $\mathfrak{T}$ has already been returned or $\mathcal{G}^{m-1} \in \mathtt{SGS}(\mathcal{P})$. Now we show that $\mathcal{G}^{m-1} \notin \mathtt{SGS}(\mathcal{P})$. The argument is basically the same as that of $m \neq 2$. That is, $\mathcal{G}^m \in \mathtt{CMC}(\mathcal{P})$ is obtained from $\mathcal{G}^{m-1}$ by an edge-deletion and thus $\mathcal{G}^{m-1} \notin \mathtt{SGS}(\mathcal{P})$. Hence, the procedure must have already returned $\mathfrak{T} = \langle \pi^1, \pi^2, ..., \pi^k \rangle$ where $|\mathfrak{T}| = k < m = |\mathfrak{C}|$. Finally, $\mathfrak{T}$ is our desired ct-sequence because π^{i+1} is a ct-mutation of π^i by construction for $1 \leq i < k$. The pairwise distinctness of $\mathcal{G}_{\pi^1}, ..., \mathcal{G}_{\pi^k}$ follows from that of $\mathcal{G}^1, ..., \mathcal{G}^m$. □

Lemma 3.5 *Given a graphoid $\mathcal{P}$, every ct-sequence $\mathfrak{T} = \langle \pi, ..., \tau \rangle$ is weakly decreasing and entails the existence of a Chickering sequence from $\mathcal{G}_\pi$ to $\mathcal{G}_\tau$. Also, $|\mathfrak{T}| \leq |\mathfrak{C}|$ for any Chickering sequence $\mathfrak{C}$ from $\mathcal{G}_\pi$ to $\mathcal{G}_\tau$.*

Proof. Every ct-sequence $\mathfrak{T} = \langle \pi, .., \tau \rangle$ is weakly decreasing due to **Lemma B.7** (c). It entails the existence of a Chickering sequence from $\mathcal{G}_\pi$ to $\mathcal{G}_\tau$ because of **Lemma B.7** (d) and **Theorem 2.6**. For the second statement, notice that $\mathcal{G}_\pi$ is either in $\mathtt{Pm}(\mathcal{P})$ or not. The case where $\mathcal{G}_\pi \notin \mathtt{Pm}(\mathcal{P})$ has already been shown in the proof of **Lemma B.8** where $|\mathfrak{T}| < |\mathfrak{C}|$ holds. For $\mathcal{G}_\pi \in \mathtt{Pm}(\mathcal{P})$, it follows from **Lemma B.7** (d) that $\mathtt{I}(\mathcal{G}_\pi) = \mathtt{I}(\mathcal{G}_\tau)$. Hence, the pairwise distinctness of DAGs induced by $\mathfrak{T}$ guarantees that each ct-mutation corresponds to a distinct covered edge reversal by **Lemma B.6**. Therefore, $|\mathfrak{T}| = |\mathfrak{C}|$ where $\mathfrak{C}$ is the shortest Chickering sequence from $\mathcal{G}_\pi$ to $\mathcal{G}_\tau$. □

Theorem 3.7 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$ and any $\pi \in \Pi(\mathbf{v})$, if $\mathcal{G}_\pi \notin \texttt{Pm}(\mathcal{P})$, then there exists a ct-sequence $\mathfrak{T} = \langle \pi, ..., \tau \rangle$ s.t. $\texttt{I}(\mathcal{G}_\pi) \subset \texttt{I}(\mathcal{G}_\tau)$ and $\mathcal{G}_\tau \in \texttt{Pm}(\mathcal{P})$.*

Proof. Immediate from **Lemma B.8**. □

Now we provide a simple example explaining how a ct-sequence $\langle \pi^1, ..., \pi^m \rangle$ is strictly shorter than any Chickering sequence from $\mathcal{G}_{\pi^1}$ to $\mathcal{G}_{\pi^m}$.

Example B.9 Consider $\mathbf{V} = \{X_1, X_2, X_3\}$ where $\texttt{I}(\mathcal{P}) = \{\langle X_1, X_2 \,|\, \varnothing \rangle\}$. Further consider $\pi^1 = \langle 3, 1, 2 \rangle$, $\pi^2 = \textit{tuck}(\pi^1, 3, 1) = \langle 1, 3, 2 \rangle$, $\pi^3 = \textit{tuck}(\pi^2, 3, 2) = \langle 1, 2, 3 \rangle$. See Figure 5 (a), (b), and (d) for their corresponding induced DAGs. Denote $\mathfrak{T}$ as the ct-sequence $\langle \pi^1, \pi^2, \pi^3 \rangle$. Observe that the shortest Chickering sequence $\mathfrak{C}$ requires three operations to transform $\mathcal{G}_{\pi^1}$ into $\mathcal{G}_{\pi^3}$ as shown in Figure 5 (a)-(d): reverse the covered $3 \to 1$, reverse the covered $3 \to 2$, and delete $1 \to 2$. In particular, $\mathcal{H}$ in Figure 5(c) is not SGS-minimal (since $\texttt{E}(\mathcal{G}_{\pi^3}) \subset \texttt{E}(\mathcal{H})$) and hence $\mathcal{H}$ is not induced by any $\pi \in \Pi(\mathbf{v})$.

(a) $\mathcal{G}_{\pi^1} = \mathcal{G}_{\langle 3,1,2 \rangle}$ (b) $\mathcal{G}_{\pi^2} = \mathcal{G}_{\langle 1,3,2 \rangle}$ (c) $\mathcal{H} \notin \texttt{SGS}(\mathcal{P})$ (d) $\mathcal{G}_{\pi^3} = \mathcal{G}_{\langle 1,2,3 \rangle}$

Figure 5: The ct-sequence $\langle \pi^1, \pi^2, \pi^3 \rangle$ is strictly shorter than the Chickering sequence $\langle \mathcal{G}_{\pi^1}, \mathcal{G}_{\pi^2}, \mathcal{H}, \mathcal{G}_{\pi^3} \rangle$.

This example above conveys three additional ideas. First, *tuck* can perform covered-edge reversal and edge-deletion in a single operation (e.g., from $\mathcal{G}_{\pi^2}$ to $\mathcal{G}_{\pi^3}$). Second, though $|\mathfrak{C}|$ and $|\mathfrak{T}|$ differ only by 1 in this example, the difference between the two kinds of sequences become greater when the initial permutation induced a DAG with multiple triangles where each can be reduced to an unshielded collider by the Chickering algorithm. Lastly, not every Chickering sequence has a ct-sequence. For instance, $\langle \mathcal{G}_{\pi^1}, \mathcal{G}_{\pi^2}, \mathcal{H} \rangle$ is a Chickering sequence. But there is no permutation inducing the non-SGS-minimal $\mathcal{H}$. This observation motivates the following lemma.

Lemma B.10 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any Chickering sequence $\mathfrak{C} = \langle \mathcal{G}^1, ..., \mathcal{G}^m \rangle$ satisfying the following:*

(i) $\texttt{I}(\mathcal{G}^1) = ... = \texttt{I}(\mathcal{G}^{m-1}) \subset \texttt{I}(\mathcal{G}^m) \subseteq \texttt{I}(\mathcal{P})$*;*

(ii) $\mathcal{G}^1, ..., \mathcal{G}^{m-2} \in \texttt{SGS}(\mathcal{P})$*, and* $\mathcal{G}^m \in \texttt{SGS}(\mathcal{P})$*;*

(iii) $\mathcal{G}^i$ *and* $\mathcal{G}^l$ *are pairwise distinct for each* $1 \leq i < l \leq m$*.*

Let π^i be a causal order of $\mathcal{G}^i$ for each $1 \leq i \leq m$. Then there exists a ct-sequence from π^1 to π^m.

Proof. Consider any $1 \leq i < m-1$. By **Lemma B.2**, we know that $\mathcal{G}^i = \mathcal{G}_{\pi^i}$ since π^i is a causal order of $\mathcal{G}^i$. Also, we know that each pair $(\mathcal{G}^i, \mathcal{G}^{i+1})$ differ by a certain covered edge reversal $(j \to k) \in \texttt{E}(\mathcal{G}^i)$ according to **Lemma B.6**. Since $\mathcal{G}^{i+1} \in \texttt{SGS}(\mathcal{P})$, it follows from **Lemma B.7** (b) that $\mathcal{G}^{i+1} = \mathcal{G}_{\pi^{i+1}} = \mathcal{G}_{\textit{tuck}(\pi^i, j, k)}$. In other words, we can express each unspecified π^{i+1} as $\textit{tuck}(\pi^i, j, k)$ accordingly for some covered edge $(j \to k) \in \texttt{E}(\mathcal{G}^i)$.

Next, consider $\mathcal{G}^{m-2} = \mathcal{G}_{\pi^{m-2}}$. Notice that $\mathcal{G}^{m-2}$ and $\mathcal{G}^{m-1}$ again differ by the reversal of some covered edge $(j \to k)$. Since $\texttt{I}(\mathcal{G}^{m-1}) \subset \texttt{I}(\mathcal{G}^m)$, $\mathcal{G}^m$ is resulted from the deletion of an edge $\texttt{E}(\mathcal{G}^{m-1})$. Now consider $\tau = \textit{tuck}(\pi^{m-2}, j, k)$. Notice that τ is a causal order of $\mathcal{G}^{m-1}$ by **Lemma B.7** (a). Since $\texttt{E}(\mathcal{G}^m) \subset \texttt{E}(\mathcal{G}^{m-1})$, τ is also a causal order of $\mathcal{G}^m$ where $\mathcal{G}_\tau = \mathcal{G}^m \in \texttt{SGS}(\mathcal{P})$ by **Lemma B.2**. Therefore, we have the desired ct-sequence $\langle \pi^1, ..., \pi^{m-2}, \tau \rangle$ where $\pi^{m-1} = \pi^m = \tau$ and the pairwise distinctness is guaranteed by (iii). □

Next, we formulate **Theorem 3.9** more formally as below.

Theorem B.11 *Given a model $(\mathcal{G}^*, \mathcal{P})$ over $\mathbf{V}$ where $\mathcal{P}$ is a graphoid and $(\mathcal{G}^*, \mathcal{P})$ satisfies a certain causal razor $\mathcal{R}$, then TSP is correct (or pointwise consistent) under $\mathcal{R}$ if and only if unbounded GRaSP$_0$ is (or pointwise consistent) correct under $\mathcal{R}$.*

Proof. The if-direction is guaranteed by **Lemma 3.5** that every ct-sequence entails a Chickering sequence. The other direction is true since the Chickering sequences examined in the DFS of TSP are exactly those satisfying the conditions in **Lemma B.10**. □

Theorem B.12 *Unbounded GRaSP$_0$ is correct and pointwise consistent under faithfulness.*

Proof. We review the argument for the correctness of unbounded GRaSP$_0$ under faithfulness given in the main paper. Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any initial permutation $\pi \in \Pi(\mathbf{v})$. Given that GRaSP$_0$ greedily search for a ct-sequence from π, it is guaranteed by **Theorem 3.7** that τ returned by unbounded GRaSP$_0$ in **Algorithm 2** induces a P-minimal DAG. Under faithfulness, we have $\mathcal{G}_\tau \in \texttt{MEC}(\mathcal{G}^*)$ due to **Theorem 2.8** (b) and hence GRaSP$_0$ is correct.

Alternatively, the correctness and pointwise consistency of unbounded GRaSP$_0$ can also be proven directly from **Theorem B.11** and the corresponding results of TSP in [Solus et al., 2021]. □

Corollary 3.8 *Unbounded GRaSP$_0$, GRaSP$_1$, and GRaSP$_2$ are correct and pointwise consistent under faithfulness.*

On the other hand, we want to prove that faithfulness is not only sufficient, but also *necessary* for the correctness of TSP and its logically equivalent unbounded GRaSP$_0$. We first want to prove an interesting equivalence between two causal razors: faithfulness and u-P-minimality.

Lemma B.13 *Given a joint probability distribution $\mathcal{P}$ over $\mathbf{V}$, for any $\langle X_i, X_j \,|\, \mathbf{X_k}\rangle \in \texttt{I}(\mathcal{P})$ for some $X_i, X_j \in \mathbf{V}$ and some $\mathbf{X_k} \subseteq \mathbf{V} \setminus \{X_i, X_j\}$, there exists $\mathcal{G} \in \texttt{DAG}(\mathbf{V})$ s.t. $\texttt{I}(\mathcal{G}) = \{\langle X_i, X_j \,|\, \mathbf{X_k}\rangle\}$.*

Proof. Consider $\mathbf{V} = \{X_1, ..., X_m\}$. An empty DAG suffices when $m = 2$. So assume that $m \geq 3$. Without loss of generality, consider $\langle X_1, X_{k+2} \,|\, \mathbf{X_k}\rangle \in \texttt{I}(\mathcal{P})$ where $\mathbf{k} = \langle 2, ..., k+1\rangle$, and the remaining vertices as $\langle k+3, ..., m\rangle$. We propose a procedure which guarantees the existence of the desired DAG $\mathcal{G}$.

---

1 $\mathcal{G} \leftarrow$ a complete undirected graph over $\mathbf{v}$
2 remove the adjacency $1 — k+2$ in $\mathcal{G}$
3 **foreach** (j, k) *that are adjacent in* $\mathcal{G}$ **do**
4 **if** $j < k$ **then**
5   orient $j \to k$ in $\mathcal{G}$
6 return $\mathcal{G}$

Line 3 to 5 guarantee that $\mathcal{G}$ is a DAG since all edges are directed and pointing from lower indices to higher indices such that no directed cycle can occur. Finally, $1 \perp_{\mathcal{G}} k+2 \,|\, \mathbf{k}$ holds because all directed paths from 1 to $k+2$ either contain a non-collider $i \in \mathbf{k}$ or contain a collider $i \notin \mathbf{k}$. Therefore, $\texttt{I}(\mathcal{G}) = \{\langle X_1, X_{k+2} \,|\, \mathbf{X_k})\}$ because no other d-separation relations hold in $\mathcal{G}$. □

Theorem 3.10 *For any joint probability distribution $\mathcal{P}$, $\texttt{CFC}(\mathcal{P}) = \texttt{uPm}(\mathcal{P})$.*

Proof. [$\subseteq$] Suppose that $\mathcal{G} \in \texttt{CFC}(\mathcal{P})$. It follows that $\mathcal{G} \in \texttt{Pm}(\mathcal{P})$ by **Definition 2.7**. For any $\mathcal{G}' \in \texttt{CMC}(\mathcal{P})$, if $\texttt{I}(\mathcal{G}') \subset \texttt{I}(\mathcal{G})$, then $\mathcal{G}' \notin \texttt{Pm}(\mathcal{P})$. Hence, if $\mathcal{G}' \in \texttt{Pm}(\mathcal{P})$, then $\texttt{I}(\mathcal{G}') = \texttt{I}(\mathcal{G})$. Hence, $\mathcal{G} \in \texttt{uPm}(\mathcal{P})$.

[$\supseteq$] Suppose that $\mathcal{G} \notin \texttt{CFC}(\mathcal{P})$. Since $\texttt{uPm}(\mathcal{P}) \subseteq \texttt{Pm}(\mathcal{P})$ by **Definition 2.7**, if $\mathcal{G} \notin \texttt{Pm}(\mathcal{P})$, we have $\mathcal{G} \notin \texttt{uPm}(\mathcal{P})$ immediately. So consider the case where $\mathcal{G} \in \texttt{Pm}(\mathcal{P})$. It follows from $\mathcal{G} \notin \texttt{CFC}(\mathcal{P})$ that there exists a CI relation $\psi \in \texttt{I}(\mathcal{P}) \setminus \texttt{I}(\mathcal{G})$. By **Lemma B.13**, we can construct a DAG $\mathcal{G}^0$ such that $\texttt{I}(\mathcal{G}^0) = \{\psi\}$. Consequently, there exists $\mathcal{G}^1 \in \texttt{Pm}(\mathcal{P})$ such that $\texttt{I}(\mathcal{G}^0) \subseteq \texttt{I}(\mathcal{G}^1) \subseteq \texttt{I}(\mathcal{P})$. Since $\psi \in \texttt{I}(\mathcal{G}^1)$, we know that $\mathcal{G}^1 \notin \texttt{MEC}(\mathcal{G})$. Given that both $\mathcal{G}, \mathcal{G}^1 \in \texttt{Pm}(\mathcal{P})$, we have $\mathcal{G} \notin \texttt{uPm}(\mathcal{P})$. □

Theorem 3.11 *Given a graphoid $\mathcal{P}$, faithfulness is necessary for the correctness of TSP and unbounded GRaSP$_0$.*

Proof. Given that TSP and unbounded GRaSP$_0$ are logically equivalent by **Theorem B.11**, we show the theorem by only considering TSP. Suppose that $(\mathcal{G}^*, \mathcal{P})$ is unfaithful. By **Theorem 3.10**, there exists $\mathcal{G} \in \texttt{Pm}(\mathcal{P})$ such that $\mathcal{G} \notin \texttt{MEC}(\mathcal{G}^*)$. Recall that Chickering algorithm can only allow us to traverse to a DAG $\mathcal{H}$ from $\mathcal{G}$ satisfying $\texttt{I}(\mathcal{G}) \subseteq \texttt{I}(\mathcal{H})$. It entails that Chickering algorithm can only obtain DAGs that are in $\texttt{MEC}(\mathcal{G})$ since $\mathcal{G} \in \texttt{Pm}(\mathcal{P})$ and hence never be able to reach $\mathcal{G}^*$ where $\texttt{I}(\mathcal{G}_\pi) \nsubseteq \texttt{I}(\mathcal{G}^*)$. Therefore, by setting π as the initial permutation to TSP where $\mathcal{G}_\pi = \mathcal{G}$, TSP will return $\mathcal{G}_\pi$ incorrectly. □

Notice that **Theorem 3.11** is contrary to what Solus et al. [2021] suggested. They proposed an example arguing that TSP can be correct even under unfaithfulness.[14] However, the distribution used in the example is not a semigraphoid. This renders their example illegitimate because every joint probability distribution is a semigraphoid.

C ESP AND GRASP-1

As shown in **Theorem 3.11** in the last section, TSP cannot be correct under unfaithfulness by choosing an arbitrary initial permutation. Consequently, one important question is how to relax the search space of TSP to identify a sparser permutation under unfaithfulness. Solus et al. [2021] proposed the *Edge SP* (ESP) algorithm based on an assumption strictly weaker than that assumed by TSP. In this section, we are going to show a theorem similar to **Theorem B.11** but with respect to ESP and unbounded GRaSP$_1$. We first examine some technical notations used in [Mohammadi et al., 2018] and [Solus et al., 2021]. Readers are strongly suggested to visit [Solus et al., 2021] for the full discussion of ESP and relevant notations.

Given a set of measured variables $\mathbf{V}$, a *permutohedron* on $\mathbf{v}$, denoted $\mathsf{A}_\mathbf{v}$, is the convex hull in $\mathbb{R}^{|\mathbf{v}|}$ of all permutations in $\Pi(\mathbf{v})$. In simpler terms, $\mathsf{A}_\mathbf{v}$ is the *state space* with each *state* being a permutation $\pi \in \Pi(\mathbf{v})$. The neighborhood of states in $\mathsf{A}_\mathbf{v}$ is defined by adjacent transpositions (ATs) as in **Definition B.1** (b). Note that different states in $\mathsf{A}_\mathbf{v}$ can induce the same DAG given a graphoid $\mathcal{P}$. Thus, a natural way to narrow down the search space is to identify permutations inducing equivalent DAGs. **Lemma B.3** provides such a characterization. Construct $\mathsf{A}_\mathbf{v}(\mathcal{P})$ by contracting neighborhood in $\mathsf{A}_\mathbf{v}$ to ATs that correspond to the CI relations in $\texttt{I}(\mathcal{P})$ specified in **Lemma B.3**. The resulted $\mathsf{A}_\mathbf{v}(\mathcal{P})$, as shown by Mohammadi et al. [2018], is a convex polytope where each vertex of $\mathsf{A}_\mathbf{v}(\mathcal{P})$ corresponds to a different induced DAG.

Nonetheless, observe that the state space in $\mathsf{A}_\mathbf{v}(\mathcal{P})$ is no longer defined in terms of $\Pi(\mathbf{v})$ because two states in $\mathsf{A}_\mathbf{v}$ inducing the same DAG are collapsed into a single state in $\mathsf{A}_\mathbf{v}(\mathcal{P})$. To draw a clearer picture, consider any $\pi, \tau \in \Pi(\mathbf{v})$ where τ is (j,k)-different from π for some $j, k \in \mathbf{v}$. They are neighbors in $\mathsf{A}_\mathbf{v}$ but they do not necessarily induce the same DAG. If $X_j \perp\!\!\!\perp_\mathcal{P} X_k \,|\, \mathbf{X}_{\texttt{Pre}(j,\pi)}$ holds, they induce the same DAG and thus correspond to the same state $\mathcal{G}_\pi$ in $\mathsf{A}_\mathbf{v}(\mathcal{P})$. But if the CI relation does not hold, then $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ are neighbors in $\mathsf{A}_\mathbf{v}(\mathcal{P})$. See Figure 6 for an example from [Solus et al., 2021].

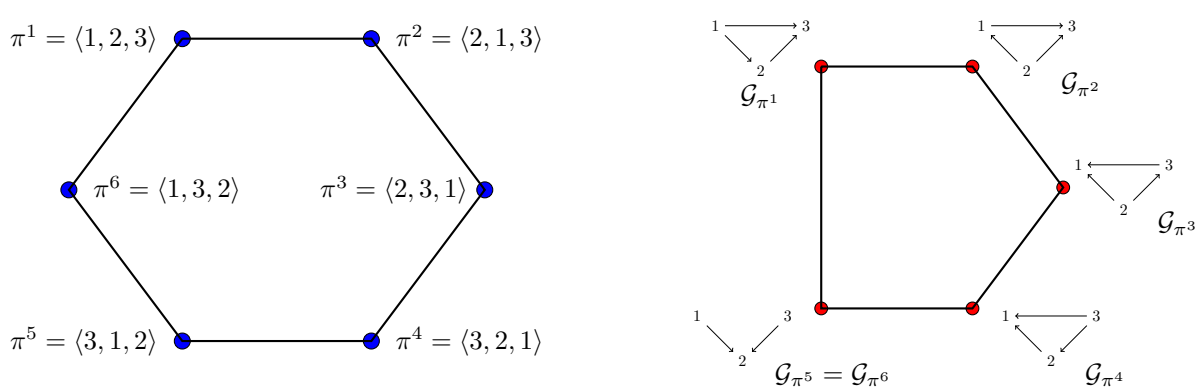


Figure 6: Given $\mathbf{V} = \{X_1, X_2, X_3\}$, consider $\texttt{I}(\mathcal{P}) = \{\langle X_1, X_3 \,|\, \varnothing \rangle\}$. The diagram on the left is the permutohedron $\mathsf{A}_\mathbf{v}$ where each state is a permutation in $\Pi(\mathbf{v})$. The one on the right is the contracted permutohedron $\mathsf{A}_\mathbf{v}(\mathcal{P})$ where each state is a different DAG in $\texttt{DAG}(\mathbf{V})$. In particular, the two states π^5 and π^6 in $\mathsf{A}_\mathbf{v}$ are collapsed into a single state in $\mathsf{A}_\mathbf{v}(\mathcal{P})$ because they induce the same DAG.

[14] See Figure 2 in the supplementary materials of [Solus et al., 2021].

Notice that each state in $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$ corresponds to a SGS-minimal DAG according to **Theorem 2.5**. ESP performs a greedy DFS in $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$. Given an initial permutation $\pi \in \Pi(\mathbf{v})$, set $\mathcal{G}$ as the induced $\mathcal{G}_\pi$ and traverse through $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$ by a *weakly decreasing walk* to obtain $\mathcal{H}$ where $|\mathtt{E}(\mathcal{H})| < |\mathtt{E}(\mathcal{G}_\pi)|$.[15] If no such $\mathcal{H}$ exists, ESP returns $\mathcal{G} = \mathcal{G}_\pi$; else $\mathcal{G}$ is reset as $\mathcal{H}$ and repeat.

As noticed by Solus et al. [2021], the construction of $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$ is inefficient since one is only required to know the neighboring states instead of the whole search space $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$ to perform the traversal. Below we show that unbounded GRaSP-1 can efficiently learn the neighbors of each state in $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$ by our permutation-based operation *tuck*. Before examining this claim, we introduce some useful definitions.

Given the (uncontracted) state space $\mathsf{A}_{\mathbf{v}}$, a *walk* $\mathfrak{W} = \langle \pi^1, ..., \pi^m \rangle$ is a sequence of neighboring states in $\mathsf{A}_{\mathbf{v}}$ such that $\pi^i, \pi^{i+1} \in \Pi(\mathbf{v})$ are in AT for each $1 \leq i < m$.

Definition C.1 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, for any walk $\mathfrak{W} = \langle \pi^1, ..., \pi^m \rangle$ in $\mathsf{A}_{\mathbf{v}}$,*

(a) $\mathfrak{W}$ is said to be DAG-preserving if $\mathcal{G}_{\pi^1} = ... = \mathcal{G}_{\pi^m}$;

(b) $\mathfrak{W}$ is said to be DAG-changing if $\langle \pi^1, ..., \pi^{m-1} \rangle$ is DAG-preserving and $\mathcal{G}_{\pi^{m-1}} \neq \mathcal{G}_{\pi^m}$.

In addition, for each DAG-changing walk $\mathfrak{W} = \langle \pi^1, ..., \pi^m \rangle$, we say that $\mathfrak{W}$ is relative to (j, k) if π^m is (j, k)-different from π^{m-1} for some $j, k \in \mathbf{v}$. Thus, each DAG-changing walk is relative to a pair of vertices corresponding to the last AT performed in the walk.

Lemma C.2 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, there exists a DAG-changing walk $\mathfrak{W} = \langle \pi^1, ..., \pi^m \rangle$ in $\mathsf{A}_{\mathbf{v}}$ if and only if $\mathcal{G}_{\pi^1}$ and $\mathcal{G}_{\pi^m}$ are neighbors in $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$.*

Proof. Given that π^{m-1} and π^m are in AT but induce different DAGs, it follows from the definition of $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$ that $\mathcal{G}_{\pi^1} = \mathcal{G}_{\pi^{m-1}}$ and $\mathcal{G}_{\pi^m}$ are neighbors in $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$. For the backward direction, suppose that $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ are neighbors in $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$ such that $\mathcal{G}_\pi \neq \mathcal{G}_\tau$. It entails that there exist $\pi', \tau' \in \Pi(\mathbf{v})$ such that π' and τ' are in AT where $\mathcal{G}_\tau = \mathcal{G}_{\tau'}$ and $\mathcal{G}_\pi = \mathcal{G}_{\pi'}$. Hence $\langle \pi', \tau' \rangle$ is our desired DAG-changing walk. □

Lemma C.3 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any pair $\pi^1, \tau^1 \in \Pi(\mathbf{v})$ such that $\pi^1 = \langle \boldsymbol{\delta}_1, j, k, \boldsymbol{\delta}_2 \rangle$ for some sub-sequences $\boldsymbol{\delta}_1$, $\boldsymbol{\delta}_2$ of π^1, and $\tau^1 = \langle \boldsymbol{\zeta}_1, j, k, \boldsymbol{\zeta}_2 \rangle$ for some sub-sequences $\boldsymbol{\zeta}_1$, $\boldsymbol{\zeta}_2$ of τ^1. Further consider $\pi^2 = \langle \boldsymbol{\delta}_1, k, j, \boldsymbol{\delta}_2 \rangle$ and $\tau^2 = \langle \boldsymbol{\zeta}_1, k, j, \boldsymbol{\zeta}_2 \rangle$. If $\mathcal{G}_{\pi^1} = \mathcal{G}_{\tau^1}$, then $\mathcal{G}_{\pi^2} = \mathcal{G}_{\tau^2}$.*

Proof. Notice that $\mathcal{G}_{\pi^2} \in \mathtt{SGS}(\mathcal{P})$ by **Theorem 2.5**. If we can show that τ^2 is a causal order of $\mathcal{G}_{\pi^2}$, it follows from **Lemma B.2** that $\mathcal{G}_{\pi^2} = \mathcal{G}_{\tau^2}$. To do so, it suffices to show the following. For any $i \in \mathbf{v} \setminus \{j, k\}$,

(i) if $(i \to j) \in \mathtt{E}(\mathcal{G}_{\pi^2})$, then $i \in \boldsymbol{\zeta}_1$;

(ii) if $(i \to k) \in \mathtt{E}(\mathcal{G}_{\pi^2})$, then $i \in \boldsymbol{\zeta}_1$;

(iii) if $(j \to i) \in \mathtt{E}(\mathcal{G}_{\pi^2})$, then $i \in \boldsymbol{\zeta}_2$;

(iv) if $(k \to i) \in \mathtt{E}(\mathcal{G}_{\pi^2})$, then $i \in \boldsymbol{\zeta}_2$.

For (i), suppose that $(i \to j) \in \mathtt{E}(\mathcal{G}_{\pi^2})$. If $(i \to j) \in \mathtt{E}(\mathcal{G}_{\pi^1})$ as well, then $(i \to j) \in \mathtt{E}(\mathcal{G}_{\tau^1})$ since $\mathcal{G}_{\pi^1} = \mathcal{G}_{\tau^1}$. This entails that $i \in \boldsymbol{\zeta}_1$. On the other hand, consider the case that $(i \to j) \notin \mathtt{E}(\mathcal{G}_{\pi^1})$. Then

$$X_i \not\perp\!\!\!\perp_{\mathcal{P}} X_j \,|\, \mathbf{X}_{\boldsymbol{\delta}_1 \setminus \{i\}} \cup \{X_k\} \qquad \because (i \to j) \in \mathtt{E}(\mathcal{G}_{\pi^2}) \tag{4}$$

$$X_i \not\perp\!\!\!\perp_{\mathcal{P}} \{X_j, X_k\} \,|\, \mathbf{X}_{\boldsymbol{\delta}_1 \setminus \{i\}} \qquad \because (4), \textit{weak union} \tag{5}$$

$$X_i \perp\!\!\!\perp_{\mathcal{P}} X_j \,|\, \mathbf{X}_{\boldsymbol{\delta}_1 \setminus \{i\}} \qquad \because (i \to j) \notin \mathtt{E}(\mathcal{G}_{\pi^1}) \tag{6}$$

$$X_i \not\perp\!\!\!\perp_{\mathcal{P}} X_k \,|\, \mathbf{X}_{\boldsymbol{\delta}_1 \setminus \{i\}} \cup \{X_j\} \qquad \because (5), (6), \textit{contraction} \tag{7}$$

By (RU), (8) entails that $(i \to k) \in \mathtt{E}(\mathcal{G}_{\pi^1}) = \mathtt{E}(\mathcal{G}_{\tau^1})$. Since τ^1 is a causal order of $\mathcal{G}_{\tau^1}$, we have $i \in \boldsymbol{\zeta}_1$.

[15] In [Solus et al., 2021], their pseudocode does not indicate that such a walk needs to be weakly decreasing but such a requirement is imposed in the description of the algorithm.

For (ii), suppose that $(i \to k) \in \mathtt{E}(\mathcal{G}_{\pi^2})$. Similar to (i), the case for $(i \to k) \in \mathtt{E}(\mathcal{G}_{\pi^1})$ is simple. So consider the case where $(i \to k) \notin \mathtt{E}(\mathcal{G}_{\pi^1})$.

$$X_i \not\perp\!\!\!\perp_{\mathcal{P}} X_k \,|\, \mathbf{X}_{\boldsymbol{\delta}_1 \setminus \{i\}} \qquad \because (i \to k) \in \mathtt{E}(\mathcal{G}_{\pi^2}) \tag{8}$$

$$X_i \not\perp\!\!\!\perp_{\mathcal{P}} \{X_j, X_k\} \,|\, \mathbf{X}_{\boldsymbol{\delta}_1 \setminus \{i\}} \qquad \because (8), \textit{decomposition} \tag{9}$$

$$X_i \perp\!\!\!\perp_{\mathcal{P}} X_k \,|\, \mathbf{X}_{\boldsymbol{\delta}_1 \setminus \{i\}} \cup \{X_j\} \qquad \because (i \to k) \notin \mathtt{E}(\mathcal{G}_{\pi^1}) \tag{10}$$

$$X_i \not\perp\!\!\!\perp_{\mathcal{P}} X_j \,|\, \mathbf{X}_{\boldsymbol{\delta}_1 \setminus \{i\}} \qquad \because (9), (10), \textit{contraction} \tag{11}$$

By (RU), (13) entails that $(i \to j) \in \mathtt{E}(\mathcal{G}_{\pi^1}) = \mathtt{E}(\mathcal{G}_{\tau^1})$ and hence $i \in \boldsymbol{\zeta}_1$.

For (iii), suppose that $(j \to i) \in \mathtt{E}(\mathcal{G}_{\pi^2})$. Then we have $(j \to i) \in \mathtt{E}(\mathcal{G}_{\pi^1})$ by (RU) because $\mathtt{Pre}(i, \pi^1) = \mathtt{Pre}(i, \pi^2)$. Hence $(j \to i) \in \mathtt{E}(\mathcal{G}_{\tau^1})$ since $\mathcal{G}_{\pi^1} = \mathcal{G}_{\tau^1}$. Given that τ^1 is a causal order of $\mathcal{G}_{\tau^1}$, we have $i \in \boldsymbol{\zeta}_2$. (iv) is analogous to (iii). □

Lemma C.4 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any two DAG-changing walks $\mathbf{W} = \langle \pi^1, ..., \pi^m \rangle$ and $\mathbf{W}' = \langle \tau^1, ..., \tau^n \rangle$ in $\mathsf{A}_{\mathbf{v}}$ where $\pi^1 = \tau^1$. If $\mathbf{W}$ and $\mathbf{W}'$ are both relative to the same (j, k) for some $j, k \in \mathbf{v}$, then $\mathcal{G}_{\pi^m} = \mathcal{G}_{\tau^n}$.*

Proof. Immediate from **Definition C.1** and **Lemma C.3**. □

Lemma C.5 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any DAG-changing walk $\mathfrak{W} = \langle \pi^1, ..., \pi^m \rangle$ in $\mathsf{A}_{\mathbf{v}}$ which is relative to (j, k) for some $j, k \in \mathbf{v}$. Then $j \to k$ is a singular edge in $\mathcal{G}_{\pi^1}$.*

Proof. Let $\mathfrak{W}_0$ denotes the DAG-preserving walk $\langle \pi^1, ..., \pi^{m-1} \rangle$. Given that π^m is (j, k)-different from π^{m-1}, it follows from **Lemma B.3** and (RU) that $(j \to k) \in \mathtt{E}(\mathcal{G}_{\pi^{m-1}})$. Since $\mathfrak{W}_0$ is a DAG-preserving walk in $\mathsf{A}_{\mathbf{v}}$, we have $(j \to k) \in \mathtt{E}(\mathcal{G}_{\pi^1}) = \mathtt{E}(\mathcal{G}_{\pi^{m-1}})$.

Next, suppose by reductio that $j \to k$ is not a singular edge in $\mathcal{G}_{\pi^1}$. Then there is a unidirectional path from j to k other than $j \to k$ in $\mathcal{G}_{\pi^1}$. So there exists $l \in \mathbf{v}$ such that $l \in \mathtt{De}(j, \mathcal{G}_{\pi^1}) \cap \mathtt{An}(k, \mathcal{G}_{\pi^1})$. In order to ensure that j and k are π^{m-1}-adjacent, either $\pi^{m-1}[l] < \pi^{m-1}[j]$ or $\pi^{m-1}[l] > \pi^{m-1}[k]$ holds. However, either case will violate that π^{m-1} is a causal order of $\mathcal{G}_{\pi^{m-1}} = \mathcal{G}_{\pi^1}$. □

Lemma C.6 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider $\pi \in \Pi(\mathbf{v})$ where $(j \to k) \in \mathtt{E}(\mathcal{G}_\pi)$ is a singular edge for some $j, k \in \mathbf{v}$. Then there exists a DAG-changing walk $\mathfrak{W} = \langle \pi, ..., \tau \rangle$ in $\mathsf{A}_{\mathbf{v}}$ relative to (j, k) where $\tau = \textit{tuck}(\pi, j, k)$.*

Proof. First, we rewrite $\pi = \langle \boldsymbol{\delta}_{<j}, j, \boldsymbol{\delta}_{j \sim k}, k, \boldsymbol{\delta}_{>k} \rangle$ as usual. Then we partition $\boldsymbol{\delta}_{j \sim k}$ as follows: $\boldsymbol{\zeta}_a = \langle i \in \boldsymbol{\delta}_{j \sim k} : i \in \mathtt{An}(k, \mathcal{G}_\pi) \rangle$, and $\boldsymbol{\zeta}_b = \langle i \in \boldsymbol{\delta}_{j \sim k} : i \notin \mathtt{An}(k, \mathcal{G}_\pi) \rangle$. Given that $(j \to k)$ is a singular edge, we know that $\mathtt{De}(j, \mathcal{G}_{\pi^1}) \cap \mathtt{An}(k, \mathcal{G}_{\pi^1}) = \varnothing$. In other words, we know that (i) each vertex in $\boldsymbol{\zeta}_a$ has no ancestor in $\boldsymbol{\delta}_{j \sim k} \setminus \boldsymbol{\zeta}_a$ in $\mathcal{G}_\pi$ and (ii) each vertex in $\boldsymbol{\zeta}_b$ has no descendant in $\boldsymbol{\delta}_{j \sim k} \setminus \boldsymbol{\zeta}_b$ in $\mathcal{G}_\pi$.

Now consider the permutation $\tau' = \langle \boldsymbol{\delta}_{<j}, \boldsymbol{\zeta}_a, j, k, \boldsymbol{\zeta}_b, \boldsymbol{\delta}_{>k} \rangle$ in particular. We want to show that there exists a DAG-preserving walk from π to τ'. Such a walk is easy to construct. First, perform repeated ATs by moving each $i \in \boldsymbol{\zeta}_a$ prior to j from left to right, and then repeated ATs by moving each $i \in \boldsymbol{\zeta}_b$ behind k from right to left. The two set of ATs are licensed by (i) and (ii) respectively. Hence, we have $\mathcal{G}_{\tau'} = \mathcal{G}_\pi$. Finally, consider $\tau = \textit{tuck}(\pi, j, k) = \langle \boldsymbol{\delta}_{<j}, \boldsymbol{\zeta}_a, k, j, \boldsymbol{\zeta}_b, \boldsymbol{\delta}_{>k} \rangle$ which is (j, k)-different from τ'. By (RU) and **Lemma B.3**, we know that $\mathcal{G}_\tau \neq \mathcal{G}_{\tau'}$ and thus $\langle \pi, ..., \tau', \tau \rangle$ is a DAG-changing walk in $\mathsf{A}_{\mathbf{v}}$ relative to (j, k). □

Theorem C.7 *Given a graphoid $\mathcal{P}$ over $\mathbf{V}$, consider any DAG-changing walk $\mathfrak{W} = \langle \pi^1, ..., \pi^m \rangle$ in $\mathsf{A}_{\mathbf{v}}$ which is relative to (j, k) for some $j, k \in \mathbf{v}$. Then $\mathcal{G}_{\pi^m} = \mathcal{G}_\tau$ where $\tau = \textit{tuck}(\pi^1, j, k)$.*

Proof. We obtain a DAG-changing walk $\mathfrak{W}' = \langle \pi^1, ..., \tau \rangle$ in $\mathsf{A}_{\mathbf{v}}$ relative to (j, k) by **Lemma C.6**. Since both $\mathfrak{W}$ and $\mathfrak{W}'$ are relative to the same (j, k), it follows from **Lemma C.4** that $\mathcal{G}_{\pi^m} = \mathcal{G}_\tau$. □

Theorem C.8 *Given a model $(\mathcal{G}^*, \mathcal{P})$ over $\mathbf{V}$ where $\mathcal{P}$ is a graphoid and $(\mathcal{G}^*, \mathcal{P})$ satisfies a certain causal razor $\mathcal{R}$, then ESP is correct (or pointwise consistent) under $\mathcal{R}$ if and only if unbounded GRaSP$_1$ is correct (or pointwise consistent) under $\mathcal{R}$.*

Proof. By **Lemma C.5** and **Lemma C.6**, every DAG-changing walk $\mathfrak{W} = \langle \pi^1, ..., \pi^m \rangle$ in $\mathsf{A}_{\mathbf{v}}$ relative to (j, k) for some $j, k \in \mathbf{v}$ corresponds to a *tuck* operation of some singular edge $j \to k$ in $\mathtt{E}(\mathcal{G}_{\pi^1})$. By **Lemma C.2**, we know that each *tuck* operation of a singular edge also corresponds to a neighboring relation between $\mathcal{G}_{\pi^1}$ and $\mathcal{G}_{\pi^m}$ in $\mathsf{A}_{\mathbf{v}}(\mathcal{P})$. Therefore, ESP and unbounded GRaSP$_1$ perform DFS over the same search space. □

D GRASP AND CAUSAL RAZORS

In this section, we discuss how a higher tier of GRaSP requires a strictly weaker assumption. As we have already seen in **Theorem B.12** and **Theorem 3.11** that faithfulness is sufficient and necessary for the correctness of unbounded GRaSP_0. In addition, we know by the construction of **Algorithm 2** that the assumption behind the correctness of unbounded GRaSP_1 (GRaSP_2) is necessarily weaker than that of unbounded GRaSP_0 (GRaSP_1). So it suffices to give examples on how a higher tier GRaSP yields a strictly sparser permutation than that from its counterpart in the lower tier. On the other hand, it is obvious that u-frugality is necessary for the correctness of unbounded GRaSP_2, but not obvious that u-frugality is insufficient. So we will also study an example with respect to this observation.

To begin with, consider how unbounded GRaSP_1 can return a strictly sparser permutation than unbounded GRaSP_0. Given the equivalence of TSP and GRaSP_0 and that of ESP and GRaSP_1 shown in **Theorem B.11** and **Theorem C.8**, we can borrow the 4-vertex example from [Solus et al., 2021] on how ESP requires a strictly weaker assumption than TSP. We refer the readers to Figure 3 in the supplementary materials of [Solus et al., 2021].

In the remainder of this section, we discuss two examples: how unbounded GRaSP_2 requires a strictly weaker assumption than unbounded GRaSP_1, and how unbounded GRaSP_2 requires a strictly stronger assumption than u-frugality. The joint distribution of each example below is a compositional graphoid. For the sake of simplicity, we only include CI relations that hold between two *singleton* sets of variables such that all other CI relations entailed by each of the graphoid properties discussed in Appendix **A.2** are understood.

Example D.1 *GRaSP_1 fails to return the sparsest permutation for some initial permutation, but GRaSP_2 succeeds in returning the sparsest permutation for every initial permutation.* Given $\mathbf{V} = \{X_1, ..., X_4\}$, consider the unfaithful model $(\mathcal{G}^*, \mathcal{P})$ where the true DAG $\mathcal{G}^*$ is shown on the left in Figure 7, and $\mathtt{I}(\mathcal{P}) = \Phi \cup \Psi$ where Φ is the set of faithful CI relations and Ψ is the set of unfaithful CI relations as listed below

$$\Phi = \{\phi_1 = \langle X_1, X_3 \,|\, \{X_2\}\rangle, \phi_2 = \langle X_2, X_4 \,|\, \{X_1, X_3\}\rangle\};$$
$$\Psi = \{\psi_1 = \langle X_2, X_4 \,|\, \varnothing\rangle\}.$$

For every $\mathcal{G} \in \mathtt{CMC}(\mathcal{P})$ where $\psi_1 \in \mathtt{I}(\mathcal{G})$, we have $5 = |\mathtt{E}(\mathcal{G})| > |\mathtt{E}(\mathcal{G}^*)| = 4$. Also, all 4-edge Markov DAGs are in the same MEC. Hence, u-frugality is satisfied. Consider feeding the initial permutation $\pi = \langle 2, 4, 1, 3\rangle$ to unbounded GRaSP_1. It will return the same π after the DFS procedure and the induced $\mathcal{G}_\pi$, as shown on the right in Figure 7, contains 5 edges. Therefore, unbounded GRaSP_1 fails to return the sparsest permutation. On the contrary, $|\mathbf{v}|! = 24$ initial permutations have been tested on unbounded GRaSP_2 and it returns $\hat{\tau}$ where $\mathcal{G}_{\hat{\tau}} \in \mathtt{MEC}(\mathcal{G}^*)$ for each initial permutation. Hence, unbounded GRaSP_2 requires an assumption strictly weaker than unbounded GRaSP_1.

Example D.2 *GRaSP_2 fails to return the sparsest permutation for some initial permutation even if u-frugality is satisfied.* This is one of the 42 examples studied in Appendix **F**. Given $\mathbf{V} = \{X_1, ..., X_5\}$, consider the unfaithful model $(\mathcal{G}^*, \mathcal{P})$ where the true DAG $\mathcal{G}^*$ is shown on the left in Figure 8, and $\mathtt{I}(\mathcal{P}) = \Phi \cup \Psi$ where Φ is the set of faithful CI relations and Ψ is the set of unfaithful CI relations as listed below:

$$\begin{aligned}\Phi = \{&\phi_1 = \langle X_1, X_2 \,|\, \varnothing\rangle, \phi_2 = \langle X_1, X_2 \,|\, \{X_3\}\rangle,\\ &\phi_3 = \langle X_2, X_3 \,|\, \varnothing\rangle, \phi_4 = \langle X_2, X_3 \,|\, \{X_1\}\rangle,\\ &\phi_5 = \langle X_2, X_5 \,|\, \{X_1, X_3, X_4\}\rangle\};\\ \Psi = \{&\psi_1 = \langle X_1, X_5 \,|\, \varnothing\rangle\}.\end{aligned}$$

For every $\mathcal{G} \in \mathtt{CMC}(\mathcal{P})$ where $\psi_1 \in \mathtt{I}(\mathcal{G})$, we have $|\mathtt{E}(\mathcal{G})| > |\mathtt{E}(\mathcal{G}^*)| = 7$. Also, all 7-edge Markov DAGs are in the same MEC. Hence, u-frugality is satisfied. Consider feeding the initial permutation $\pi = \langle 5, 1, 3, 4, 2\rangle$ to unbounded GRaSP_2. It will return the same π after the DFS procedure and the induced $\mathcal{G}_\pi$, as shown on the right in Figure 8, contains 8 edges. Therefore, unbounded GRaSP_2 fails to return the sparsest permutation even under u-frugality.

Corollary D.3 *Given a graphoid $\mathcal{P}$, unbounded GRaSP_2 is correct under a strictly weaker assumption than unbounded GRaSP_1 (and ESP), and unbounded GRaSP_1 (and ESP) is correct under a strictly weaker assumption than unbounded GRaSP_0 (and TSP).*

Figure 7: An unfaithful model satisfying u-frugality. The true DAG $\mathcal{G}^*$ is shown on the left where the two shaded vertices indicate the unfaithful marginal independence $X_2 \perp\!\!\!\perp_{\mathcal{P}} X_4 \mid \varnothing$. Unbounded GRaSP$_1$ returns its initial permutation $\pi = \langle 2, 4, 1, 3 \rangle$. The induced DAG $\mathcal{G}_\pi$ is shown on the right with 5 edges. However, unbounded GRaSP$_2$ manages to return a sparsest permutation under every initial permutation.

Figure 8: An unfaithful model satisfying u-frugality. The true DAG $\mathcal{G}^*$ is shown on the left where the two shaded vertices indicate the unfaithful marginal independence $X_1 \perp\!\!\!\perp_{\mathcal{P}} X_5 \mid \varnothing$. Unbounded GRaSP$_2$ returns its initial permutation $\pi = \langle 5, 1, 3, 4, 2 \rangle$. The induced DAG $\mathcal{G}_\pi$ is shown on the right with 8 edges. Hence, GRaSP$_2$ is not correct under u-frugality alone.

E GROW-SHRINK ALGORITHM AND ITS PROPERTIES

Definition E.1 *Given an observational dataset $\mathcal{D}$ with n i.i.d. observations from a joint probability distribution $\mathcal{P}$ over $\mathbf{V}$ that belongs to a curved exponential family*[16]*, for every $X \in \mathbf{V}$ and every $\mathbf{M} \subseteq \mathbf{V} \setminus X$,*

$$\texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}) = \ell_{X|\mathbf{M}}(\hat{\theta}_{\texttt{mle}} \mid \mathcal{D}) + c\,\frac{|\hat{\theta}_{\texttt{mle}}|}{2}\log(n)$$

where $\ell_{X|\mathbf{M}}$ is the conditional log likelihood function, $|\hat{\theta}_{\texttt{mle}}|$ is the absolute value of the maximum likelihood estimate, and c is a multiplier for the parameter penalty.

BIC score is a *decomposable* scoring function in the sense that the BIC score of any DAG $\mathcal{G}$ (over the same set of variables $\mathbf{V}$ as the observational dataset $\mathcal{D}$), denoted as $\texttt{BIC}_{\mathcal{D}}(\mathcal{G})$, satisfies the following:

$$\texttt{BIC}_{\mathcal{D}}(\mathcal{G}) = \sum_{i \in \mathbf{v}} \texttt{BIC}_{\mathcal{D}}(X_i, \mathbf{X}_{\texttt{Pa}(i,\mathcal{G})}).$$

In addition, since we will be using BIC throughout this appendix, we assume that every joint probability distribution $\mathcal{P}$ belongs to a curved exponential family in this section.

[16]See [Kass and Vos, 2011] for an in-depth analysis of curved exponential families.

Algorithm 3: GROW: *grow*$(\mathcal{D}, X, \mathbf{Z})$

Input: (a) $\mathcal{D}$: an observational dataset over $\mathbf{V}$; (b) $X \in \mathbf{V}$; (c) $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X\}$
Output: $\mathbf{M}_{gr} \subseteq \mathbf{Z}$

$s \leftarrow \texttt{BIC}_{\mathcal{D}}(X, \varnothing)$
$s' \leftarrow s$
$\mathbf{M}_{gr} \leftarrow \varnothing$
do
  $s \leftarrow s'$
  $s' \leftarrow \max_{Y \in \mathbf{Z} \setminus \mathbf{M}_{gr}} \texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}_{gr} \cup \{Y\})$
  $Y' \leftarrow \text{argmax}_{Y \in \mathbf{Z} \setminus \mathbf{M}_{gr}} \texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}_{gr} \cup \{Y\})$
  if $s' > s$ **then**
    $\mathbf{M}_{gr} \leftarrow \mathbf{M}_{gr} \cup \{Y'\}$
while $s' > s$
return $\mathbf{M}_{gr}$

Algorithm 4: SHRINK: *shrink*$(\mathcal{D}, X, \mathbf{Z})$

Input: (a) $\mathcal{D}$: an observational dataset over $\mathbf{V}$; (b) $X \in \mathbf{V}$; (c) $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X\}$
Output: (i) $\mathbf{M}_{sh} \subseteq \mathbf{Z}$; (ii) $s = \texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}_{sh})$

$s \leftarrow \texttt{BIC}_{\mathcal{D}}(X, \mathbf{Z})$
$s' \leftarrow s$
$\mathbf{M}_{sh} \leftarrow \mathbf{Z}$
do
  $s \leftarrow s'$
  $s' \leftarrow \max_{Y \in \mathbf{M}_{sh}} \texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}_{sh} \setminus \{Y\})$
  $Y' \leftarrow \text{argmax}_{Y \in \mathbf{M}_{sh}} \texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}_{gs} \setminus \{Y\})$
  if $s' > s$ **then**
    $\mathbf{M}_{sh} \leftarrow \mathbf{M}_{sh} \setminus \{Y'\}$
while $s' > s$
return $\mathbf{M}_{sh}, s$

Theorem E.2 *[Chickering, 2002] Given an observational dataset $\mathcal{D}$ with n i.i.d. observations from a joint probability distribution $\mathcal{P}$ over $\mathbf{V}$, consider $\mathcal{G}, \mathcal{G}' \in \texttt{DAG}(\mathbf{V})$ where $\mathcal{G}'$ is resulted from adding the edge $j \to k$ in $\mathcal{G}$. In the large sample limit of n,*

(a) if $X_j \not\perp\!\!\!\perp_{\mathcal{P}} X_k \mid \mathbf{X}_{\texttt{Pa}(k,\mathcal{G})}$, then $\texttt{BIC}_{\mathcal{D}}(\mathcal{G}') > \texttt{BIC}_{\mathcal{D}}(\mathcal{G})$;
(b) if $X_j \perp\!\!\!\perp_{\mathcal{P}} X_k \mid \mathbf{X}_{\texttt{Pa}(k,\mathcal{G})}$, then $\texttt{BIC}_{\mathcal{D}}(\mathcal{G}') < \texttt{BIC}_{\mathcal{D}}(\mathcal{G})$.

The theorem above is known as the *local consistency* of BIC score over DAGs. We can easily derive a lemma which concerns the BIC score of a variable (relative to a set of variables).

Lemma E.3 *Given an observational dataset $\mathcal{D}$ with n i.i.d. observations from a joint probability distribution $\mathcal{P}$ over $\mathbf{V}$, consider any distinct $j, k \in \mathbf{v}$ and $\mathbf{i} \subseteq \mathbf{v} \setminus \{j, k\}$. In the large sample limit of n,*

(a) if $X_j \not\perp\!\!\!\perp_{\mathcal{P}} X_k \mid \mathbf{X}_{\mathbf{i}}$, then $\texttt{BIC}_{\mathcal{D}}(X_k, \mathbf{X}_{\mathbf{i}} \cup \{X_j\}) > \texttt{BIC}_{\mathcal{D}}(X_k, \mathbf{X}_{\mathbf{i}})$;
(b) if $X_j \perp\!\!\!\perp_{\mathcal{P}} X_k \mid \mathbf{X}_{\mathbf{i}}$, then $\texttt{BIC}_{\mathcal{D}}(X_k, \mathbf{X}_{\mathbf{i}} \cup \{X_j\}) < \texttt{BIC}_{\mathcal{D}}(X_k, \mathbf{X}_{\mathbf{i}})$.

Proof. Construct a DAG $\mathcal{G} \in \texttt{DAG}(\mathbf{V})$ by drawing all and only directed edges from each vertex in $\mathbf{i}$ to k, and another DAG $\mathcal{G}' \in \texttt{DAG}(\mathbf{V})$ by adding $j \to k$ in $\mathcal{G}$. Then the lemma immediately follows from **Theorem E.2** and the decomposable feature of BIC scores. □

Lemma E.4 *Consider an observational dataset $\mathcal{D}$ with n i.i.d. observations from a compositional graphoid $\mathcal{P}$ over $\mathbf{V}$. In the large sample limit of n, for any $X \in \mathbf{V}$ and any $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X\}$, $\texttt{MB}(X, \mathbf{Z}) \subseteq \mathbf{M}_{gr}$ where $\mathbf{M}_{gr} = grow(\mathcal{D}, X, \mathbf{Z}) \subseteq \mathbf{Z}$.*

Proof. First, **Algorithm 3** requires that $\mathbf{M}_{gr} \subseteq \mathbf{Z}$, and $\texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}_{gr} \cup \{Y\}) < \texttt{BIC}_{D}(X, \mathbf{M}_{gr})$ for every $Y \in \mathbf{Z} \setminus \mathbf{M}_{gr}$. By **Lemma E.3**, we have $X \perp\!\!\!\perp_{\mathcal{P}} Y \,|\, \mathbf{M}_{gr}$ for each $Y \in \mathbf{Z} \setminus \mathbf{M}_{gr}$. By composition, we have $X \perp\!\!\!\perp_{\mathcal{P}} (\mathbf{Z} \setminus \mathbf{M}_{gr}) \,|\, \mathbf{M}_{gr}$. Therefore, by **Definition A.1** and **Lemma A.2**, we have $\texttt{MB}(X, \mathbf{Z}) \subseteq \mathbf{M}_{gr}$. □

Lemma E.5 *Consider an observational dataset $\mathcal{D}$ with n i.i.d. observations from a graphoid $\mathcal{P}$ over $\mathbf{V}$. In the large sample limit of n, for any $X \in \mathbf{V}$ and any $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X\}$, $\texttt{MB}(X, \mathbf{Z}) = \mathbf{M}_{sh}$ where $\mathbf{M}_{sh} = \textit{shrink}(\mathcal{D}, X, \mathbf{Z}) \subseteq \mathbf{Z}$.*

Proof. We show the lemma by $\mathbf{M}_{sh} \subseteq \texttt{MB}(X, \mathbf{Z})$ and $\mathbf{M}_{sh} \supseteq \texttt{MB}(X, \mathbf{Z})$.

[$\subseteq$] By reductio, suppose that there exists $Y \in \mathbf{M}_{sh} \subseteq \mathbf{Z}$ but $Y \notin \texttt{MB}(X, \mathbf{Z})$. Let $\mathbf{S}$ be $\mathbf{M}_{sh} \setminus \{Y\}$. **Algorithm 4** requires that $\texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}_{sh} \setminus \{Y\}) < \texttt{BIC}_{D}(X, \mathbf{M}_{sh})$. In other words, we have $\texttt{BIC}_{D}(X, \mathbf{S}) < \texttt{BIC}_{\mathcal{D}}(X, \mathbf{S} \cup \{Y\})$. By **Lemma E.3**, we have $X \not\perp\!\!\!\perp_{\mathcal{P}} Y \,|\, \mathbf{S}$.

Let $\mathbf{W} = \mathbf{S} \setminus \texttt{MB}(X, \mathbf{Z})$. From $Y \notin \texttt{MB}(X, \mathbf{Z})$ and $Y \notin \mathbf{S}$, we have $\{Y\} \cup \mathbf{W} \subseteq \mathbf{Z} \setminus \texttt{MB}(X, \mathbf{Z})$. Recall **Definition A.1** that $X \perp\!\!\!\perp_{\mathcal{P}} \mathbf{Z} \setminus \texttt{MB}(X, \mathbf{Z}) \,|\, \texttt{MB}(X, \mathbf{Z})$. Thus,

$$X \perp\!\!\!\perp_{\mathcal{P}} \{Y\} \cup \mathbf{W} \,|\, \texttt{MB}(X, \mathbf{Z}) \qquad \because X \perp\!\!\!\perp_{\mathcal{P}} \mathbf{Z} \setminus \texttt{MB}(X, \mathbf{Z}) \,|\, \texttt{MB}(X, \mathbf{Z}),\ \textit{decomposition} \tag{12}$$

$$X \perp\!\!\!\perp_{\mathcal{P}} Y \,|\, \texttt{MB}(X, \mathbf{Z}) \cup \mathbf{W} \qquad \because (12),\ \textit{weak union} \tag{13}$$

$$X \perp\!\!\!\perp_{\mathcal{P}} Y \,|\, \mathbf{S} \qquad \because (13), \mathbf{W} = \mathbf{S} \setminus \texttt{MB}(X, \mathbf{Z}) \tag{14}$$

Contradiction arises with $X \not\perp\!\!\!\perp_{\mathcal{P}} Y \,|\, \mathbf{S}$.

[$\supseteq$] Observe that **Algorithm 4** removes one variable in $\mathbf{Z}$ one at a time repeatedly to form $\mathbf{M}_{sh}$. Thus, the shrink-procedure corresponds to a sequence of sets of variables $\langle \mathbf{M}^0, ..., \mathbf{M}^k \rangle$ and a sequence of variables $\mathbf{W} = \langle W_1, ..., W_k \rangle = \mathbf{Z} \setminus \mathbf{M}_{sh}$ such that $\mathbf{M}^0 = \mathbf{Z}$, $\mathbf{M}^k = \mathbf{M}_{sh}$, and $\mathbf{M}^i = \mathbf{M}^{i-1} \setminus \{W_i\}$ (where $W_i \in \mathbf{M}^{i-1}$) for each $1 < i \leq k$.

Notice that $\mathbf{M}^{i-1} = \mathbf{M}^i \cup \{W_i\}$. **Algorithm 4** requires that $\texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}^i) > \texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}^{i-1}) = \texttt{BIC}_{D}(X, \mathbf{M}^i \cup \{W_i\})$. We then have

$$X \perp\!\!\!\perp_{\mathcal{P}} W_1 \,|\, \mathbf{M}^1 \qquad \because \texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}^1) > \texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}^0), \textbf{Lemma E.3} \tag{15}$$

$$X \perp\!\!\!\perp_{\mathcal{P}} W_2 \,|\, \mathbf{M}^2 \qquad \because \texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}^2) > \texttt{BIC}_{\mathcal{D}}(X, \mathbf{M}^1), \textbf{Lemma E.3} \tag{16}$$

$$X \perp\!\!\!\perp_{\mathcal{P}} W_1 \,|\, \mathbf{M}^2 \cup \{W_2\} \qquad \because (15), \mathbf{M}^1 = \mathbf{M}^2 \cup \{W_2\} \tag{17}$$

$$X \perp\!\!\!\perp_{\mathcal{P}} \{W_1, W_2\} \,|\, \mathbf{M}^2 \qquad \because (16), (17), \textit{contraction} \tag{18}$$

$$\vdots$$

$$X \perp\!\!\!\perp_{\mathcal{P}} \{W_1, ..., W_k\} \,|\, \mathbf{M}^k \qquad \because ..., \textit{contraction} \tag{19}$$

$$X \perp\!\!\!\perp_{\mathcal{P}} \mathbf{W} \,|\, \mathbf{M}_{sh} \qquad \because (19), \mathbf{W} = \langle W_1, ..., W_k \rangle \text{ and } \mathbf{M}^k = \mathbf{M}_{sh} \tag{20}$$

$$X \perp\!\!\!\perp_{\mathcal{P}} \mathbf{Z} \setminus \mathbf{M}_{sh} \,|\, \mathbf{M}_{sh} \qquad \because (20), \mathbf{W} = \mathbf{Z} \setminus \mathbf{M}_{sh} \tag{21}$$

Hence, it follows from **Definition A.1** that $\mathbf{M}_{sh} \supseteq \texttt{MB}(X, \mathbf{Z})$. □

Theorem E.6 *Consider an observational dataset $\mathcal{D}$ with n i.i.d. observations from a compositional graphoid $\mathcal{P}$ over $\mathbf{V}$. In the large sample limit of n, for any $X \in \mathbf{V}$ and any $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X\}$, $\texttt{MB}(X, \mathbf{Z}) = \mathbf{M}_{gs}$ where $\mathbf{M}_{gs} = \textit{shrink}(\mathcal{D}, X, \textit{grow}(\mathcal{D}, X, \mathbf{Z}))$.*

Proof. Immediate from **Lemma E.4** and **Lemma E.5**. □

Theorem 3.12 is simply the conjunction of **Lemma E.5** and **Theorem E.6**. Lastly, we show how the score returned by *shrink* is related to the BIC score of a DAG.

Theorem E.7 *Consider an observational dataset $\mathcal{D}$ with n i.i.d. observations from a (compositional) graphoid $\mathcal{P}$ over $\mathbf{V} = \{X_1, ..., X_m\}$, and any $\pi \in \Pi(\mathbf{v})$. Let s_i and $\mathbf{M}_i$ be the score and the set of variables returned by $\textit{shrink}(\mathcal{D}, X_i, \mathbf{X}_{\texttt{Pre}(i,\pi)})$ (or $\textit{shrink}(\mathcal{D}, X_i, \textit{grow}(\mathcal{D}, X_i, \mathbf{X}_{\texttt{Pre}(i,\pi)}))$ if $\mathcal{P}$ is a compositional graphoid) respectively. Denote s_π as $\sum_{i \in \mathbf{v}} s_i$. In the large sample limit of n, $\texttt{BIC}_{\mathcal{D}}(\mathcal{G}_\pi) = s_\pi$ where $\mathcal{G}_\pi$ is induced from π by (VP).*

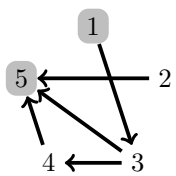
Proof. Immediate from the decomposable feature of BIC scores and **Lemma E.5** (or **Theorem 3.12**). □

Lastly, though a compositional graphoid is a sufficient condition for the correct identification of the unique Markov boundary using the grow-shrink algorithm, we are aware of an assumption weaker than compositional graphoid to validate such an identification. Nevertheless, this discussion will be beyond the scope of this paper and we will leave the formal proof to future work.

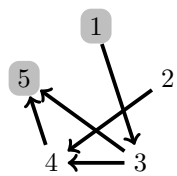
F UNIT TESTS

We are actively investigating a rigorous treatment of path cancellation in DAG models. Accordingly, we acknowledge that our current formulation, and therefore the derived models below, could be erroneous. However, the models listed in this appendix are compositional graphoids that are unfaithful and u-frugal with respect to DAGs. As such, even if they do not correspond to DAG models with path cancelations, they are still interesting examples to investigate.

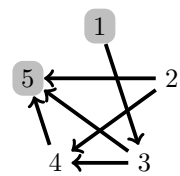
Assuming our formulation is correct, we consider path cancellations in DAGs between vertices connected by two or more unique treks. Furthermore, the path cancellations we consider elicit a marginal independence between the two variables in question. Below, we enumerate all possible path cancellations of this type (up to vertex relabeling) between an exogenous variable and an endogenous variable. Each graph illustrates a case where an unfaithful marginal independence, the first in the list, is elicited between the two gray vertices due to path cancellation. The other CI relations in the list are entailed by the graphoid properties, is symmetry assumed.



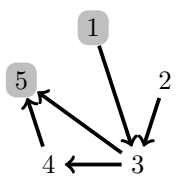
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



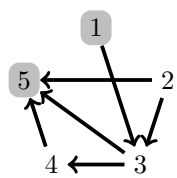
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



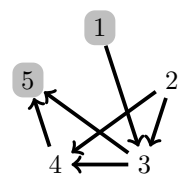
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



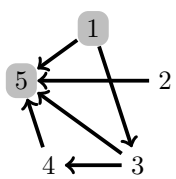
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



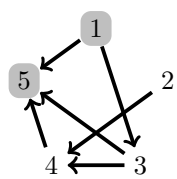
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



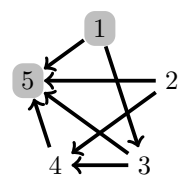
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



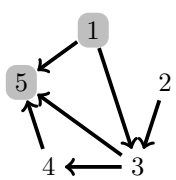
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



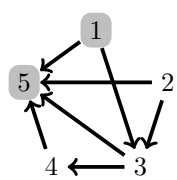
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



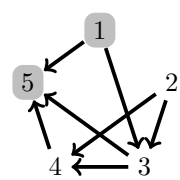
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



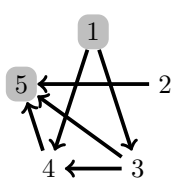
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



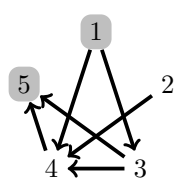
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



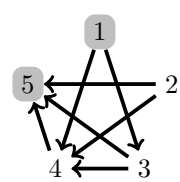
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



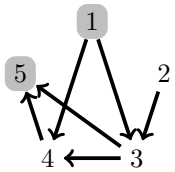
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



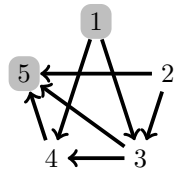
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



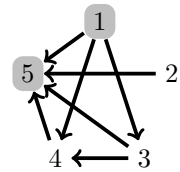
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



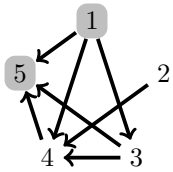
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



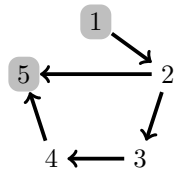
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



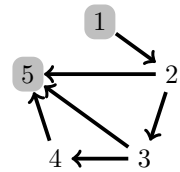
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



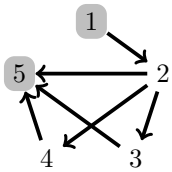
$X_1 \perp\!\!\!\perp X_5$
$X_1 \perp\!\!\!\perp X_2, X_5$
$X_1 \perp\!\!\!\perp X_2 \mid X_5$
$X_1 \perp\!\!\!\perp X_5 \mid X_2$



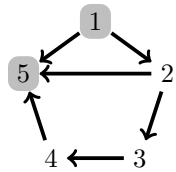
$X_1 \perp\!\!\!\perp X_5$



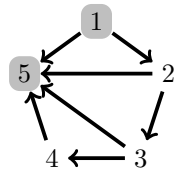
$X_1 \perp\!\!\!\perp X_5$



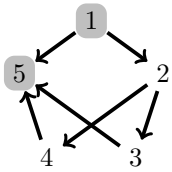
$X_1 \perp\!\!\!\perp X_5$



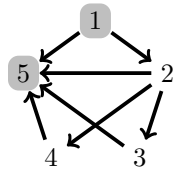
$X_1 \perp\!\!\!\perp X_5$



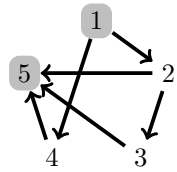
$X_1 \perp\!\!\!\perp X_5$



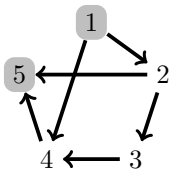
$X_1 \perp\!\!\!\perp X_5$



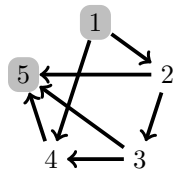
$X_1 \perp\!\!\!\perp X_5$



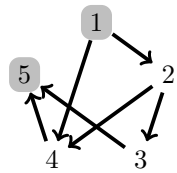
$X_1 \perp\!\!\!\perp X_5$



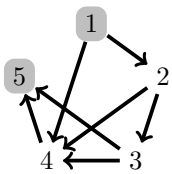
$X_1 \perp\!\!\!\perp X_5$



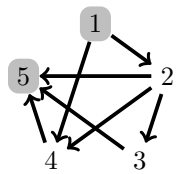
$X_1 \perp\!\!\!\perp X_5$



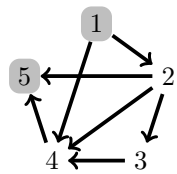
$X_1 \perp\!\!\!\perp X_5$



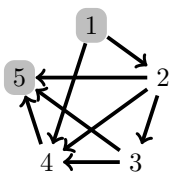
$X_1 \perp\!\!\!\perp X_5$



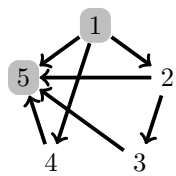
$X_1 \perp\!\!\!\perp X_5$



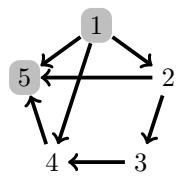
$X_1 \perp\!\!\!\perp X_5$



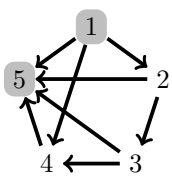
$X_1 \perp\!\!\!\perp X_5$



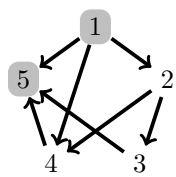
$X_1 \perp\!\!\!\perp X_5$



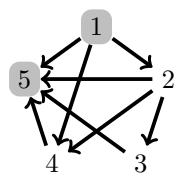
$X_1 \perp\!\!\!\perp X_5$



$X_1 \perp\!\!\!\perp X_5$



$X_1 \perp\!\!\!\perp X_5$

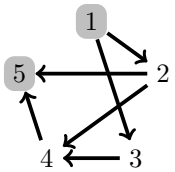


$X_1 \perp\!\!\!\perp X_5$

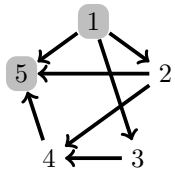
$X_1 \perp\!\!\!\perp X_5$

$X_1 \perp\!\!\!\perp X_5$

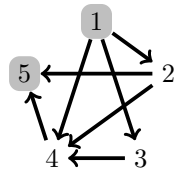
$X_1 \perp\!\!\!\perp X_5$



$X_1 \perp\!\!\!\perp X_5$



$X_1 \perp\!\!\!\perp X_5$



$X_1 \perp\!\!\!\perp X_5$