Revisiting Causal Discovery from a Complexity-Theoretic Perspective

Robert Ganian , **Viktoriia Korchemna** , **Stefan Szeider**
Algorithms and Complexity Group, TU Wien, Vienna, Austria
{rganian,vkorchemna,sz}@ac.tuwien.ac.at

Abstract

Causal discovery seeks to unveil causal relationships (represented as a so-called causal graph) from observational data. This paper investigates the complex relationship between the graph structure and the efficiency of constraint-based causal discovery algorithms.

Our main contributions include (i) a near-tight characterization of which causal graphs admit a small d-separating set for each pair of vertices and thus can potentially be efficiently recovered by a constraint-based causal discovery algorithm, (ii) the explicit construction of a sequence of causal graphs on which the influential PC algorithm might need exponential time, although there is a small d-separating set between every pair of variables, and (iii) the formulation of a new causal discovery algorithm which achieves fixed-parameter running time by considering the maximum number of edge-disjoint paths between variables in the (undirected) super-structure as the parameter.

A distinguishing feature of our investigation is that it is carried out within a more fine-grained model which more faithfully captures the infeasibility of performing accurate independence tests for large sets of conditioning variables.

1 Introduction

Causality has become a crucial topic in AI research for overcoming the limitations of Machine Learning systems that are typically based on correlation. While *causal inference* focuses on estimating the effect of a known or hypothesized causal relationship, *causal discovery* (the subject of this paper) aims to uncover new causal relationships directly from data, often without prior hypotheses.

The causal structure over a set of random variables can be explicitly expressed as a *causal graph*, which is a directed graph whose vertices are the variables, and where an arc $X \to Y$ indicates that X is a direct cause of Y. Causal discovery is the problem of identifying as much as possible about the causal graph given a dataset of measurements over the underlying variables; in some cases, part of the causal relationships among the variables is provided from expert knowledge [Ng *et al.*, 2021; Chen *et al.*, 2016; Tsamardinos *et al.*, 2006; Ramaswamy and Szeider, 2022] which can be presented in terms of a *super-structure* [Perrier *et al.*, 2008; Ordyniak and Szeider, 2013; Ganian and Korchemna, 2021; Grüttemeier and Komusiewicz, 2022], a graph containing all edges which are known or allowed to be part of the causal graph in some orientation.

An important family of algorithms for causal discovery are *constraint-based algorithms* which work with a (usually very large) set of observational data and repeatedly perform independence checks between pairs of variables, i.e., test $X \perp Y \mid Z$. Here, it is crucial to distinguish how one models these tests: one commonly assumes these are performed by an oracle which can either (1) determine whether X and Y are conditionally independent via a single query, or (2) test independence under a specific assignment of the conditioning set Z. Here, we take the latter approach—which we from now on refer to as the *fine-grained oracle model*—as it allows us to more faithfully model settings where, e.g., the data needs to be obtained via on-the-fly experiments or when the amount of data is excessively large. We note that under the fine-grained oracle model, performing independence tests for very large choices of Z becomes infeasible—a fact which also corresponds well to the known unreliability of such independence tests [Wienöbst and Liskiewicz, 2021; Talvitie and Parviainen, 2020].

The *PC algorithm*, proposed by Peter and Clarke [Spirtes *et al.*, 2000], is a constraint-based algorithm that had a monumental impact on causal discovery and served as the basis for several other constraint-based algorithms; in fact, it is considered the "default algorithm for attempting causal discovery" [Shalizi, 2023]. The statistical independence check used by the PC algorithm (and many other constraint-based causal discovery algorithms) is based on so-called d-separating sets. Since the running time of the test in the fine-grained oracle model is exponential in the size of the d-separating set, the PC algorithm performs the tests with gradually increasing d-separating sets, starting from checking unconditional independence (which corresponds to checking for the existence of an empty d-separating set).

The PC and other constraint-based algorithms proposed for causal discovery over the last two decades have been refined and analyzed from statistical and empirical perspectives

(see the discussion of related work at the end of the Introduction). However, relatively little is known about the complexity-theoretic foundations of constraint-based causal discovery algorithms in the fine-grained oracle model. We perform the first complexity-theoretic analysis in this more refined setting by attacking the following three fundamental research questions.

Q1 *Which structural properties of the causal graph guarantee small d-separating sets between any pair of variables?* A small d-separating set is a prerequisite for applying constraint-based causal discovery algorithms efficiently and provides guarantees for their running times.
Q2 *How large are the d-separating sets discovered by the PC algorithm compared to smallest d-separating sets?* Let us refer to the difference between the size of a d-separating set identified by the PC algorithm and the size of a smallest such set as the "approximation error". The smaller the approximation error, the closer the PC algorithm's time complexity is to the time complexity of an idealistic constraint-based algorithm.
Q3 *Which properties of the provided super-structure guarantee efficient causal discovery?* We consider this question with respect to the PC algorithm, but also for new constraint-based algorithms that we design to exploit the properties of the super-structure better.

1.1 Results

Addressing Q1: We obtain a near-tight characterization of which causal graphs admit a small d-separating set for each pair of vertices. In particular, if the number of edge-disjoint paths between each pair of vertices in the skeleton of the causal graph (i.e., the undirected graph underlying the causal graph) is bounded, then there is a d-separating set between each pair of vertices of bounded size (Theorem 3). At the same time, the statement cannot be strengthened to simply speak of edge-disjoint directed paths (Observation 4) or to the number of vertex-disjoint paths in the skeleton (Theorem 5). Conversely, if a causal graph admits a d-separating set of size at most ℓ for each pair of non-adjacent vertices, then for each pair of non-adjacent vertices, the number of directed vertex-disjoint paths is bounded by the same bound ℓ (Observation 6, Corollary 7). We show that this result is tight in the sense that it cannot be improved to a bound on the number of vertex-disjoint or edge-disjoint paths in the skeleton (Lemma 8) or to a bound on the number of directed edge-disjoint paths in the causal graph (Lemma 9). These results are presented in Section 3.

Addressing Q2: We first observe that no constraint-based causal discovery algorithm can run faster than in time n^k, where n is the number of variables and k is the size of a smallest d-separating set between any pair of variables (Lemma 10). In other words, the size of a smallest d-separating set is a hard complexity bound for all constraint-based algorithms.

In the literature, it has been claimed that the PC algorithm will find and use a smallest d-separating set between every pair of variables [Claassen *et al.*, 2013, Subsection 3.1]. We, in fact, show that this is not the case: the PC algorithm might fail to run efficiently due to failing to find a small d-separating set even when such a set exists. This would be relatively easy to show when expert knowledge is provided, but our construction works even in the base case with no expert knowledge (i.e., when the super-structure is a complete graph). In particular, we construct a non-trivial class of instances where the PC algorithm only discovers a d-separating set of size linear in the total number of variables, even though there exists a d-separating set of size 3 (Theorem 11). This direction is investigated within our Section 4.

Addressing Q3: We have now seen two obstacles to efficient causal discovery: if the d-separating set is large, then no constraint-based algorithm will run efficiently, but even if it is small, the PC algorithm might not run efficiently. Here, we consider whether natural graph-theoretic structural restrictions on the super-structure can help us address both of these obstacles to guarantee efficient causal discovery for cases where some (suitable) expert knowledge is available. In particular, we aim to avoid the exponential blowup of n^k by designing so-called *fixed-parameter algorithms* for the problem.

A typical restriction one would consider in the setting of fixed-parameter tractability would be to consider super-structures of bounded treewidth [Robertson and Seymour, 1986]; however, as our first result in this direction we prove that using treewidth (as well as many other more restrictive variants of it that have been used in this context) as a parameter cannot yield fixed-parameter tractability for causal discovery, for any constraint-based algorithm (Corollary 12). On the other hand, we show that the maximum degree in the super-structure is a parameter which guarantees that the PC algorithm will terminate in fixed-parameter time (Theorem 13). Unfortunately, the restriction to super-structures of bounded degree only is very strong, and so this result begs the question of whether tractability can be extended to super-structures of unbounded degree.

We show that by a slight modification of the PC algorithm, we can lift the tractability for the bounded-degree case to parameterizing by the maximum degree within each 2-connected component (Theorem 14); it is worth noting that fixed-parameter tractability for the PC algorithm without our modification cannot be guaranteed in this case (Corollary 15). We conclude by developing a new causal discovery algorithm that guarantees a fixed-parameter running time in a much more general setting—in particular, there it suffices to parameterize by the maximum number of edge-disjoint paths between any pair of variables (Theorem 16)—and again match this with a lower bound showing that the PC algorithm (even the modified one) cannot solve this case efficiently (Corollary 17). This third set of results is presented in Section 5.

1.2 Related Work

There are two main approaches to causal discovery: constraint-based and score-based. The complexity of constraint-based causal discovery is well-studied in terms of the number of conditional independence tests. As Claassen *et al.* [2013] noted, the PC algorithm allows learning causal networks whose node degree is upper-bounded by k via at most $n^{2(k+2)}$ independence tests. Recent works have also explored other constraint-based causal discovery algorithms, including the refinement ED-PC of the PC algorithm by Wienöbst and Liskiewicz [2021] and MARVEL [Mokhtarian *et al.*, 2021]; the latter achieves nearly-tight bounds in the number of independence

tests performed. However, these results do not carry over to the fine-grained oracle model since the algorithms assume the possibility of performing efficient independence tests with large conditioning sets; the sizes of these sets are only upper-bounded by the maximal in-degree.

The importance of reducing the sizes of conditioning sets was also mentioned in many other articles (see, e.g., the discussion in the work of Mokhtarian *et al.* [2022]). For instance, Talvitie and Parviainen [2020] aimed to bound the number of oracle calls and the size of d-separating sets. They did so by considering the so-called *moral graph* of the hidden causal graph and showed that whenever the moral graph has treewidth at most k, each pair of variables admits a d-separating set of size at most $k + 1$. This is incomparable to our results addressing Q1: we show that the size of a d-separating set is upper-bounded by a function of the maximum edge-cut of a skeleton of the hidden causal graph (which allows us to obtain strong bounds even for skeletons with high-degree nodes), but it is easy to observe that a bound on the treewidth of the moral graph implies a bound on the degree of the skeleton. An example of an instance where our bounds supersede previous results can be seen, e.g., in Figure 4.

It is known that the smallest d-separating set between X and Y corresponds to the smallest vertex separator in a moral graph of D restricted to ancestors of X and Y [Tian *et al.*, 1998]. We show that this characterization cannot be extended to skeletons (and, therefore, neither to the super-structure setting) by constructing a family of networks with skeletons containing a vertex separator of size 3 between every pair of variables but arbitrarily large smallest d-separating sets between some pair of variables (Theorem 5).

Under the Faithfulness and Causal Markov conditions, causal networks are known to be Bayesian [Mokhtarian *et al.*, 2022]. The parameterized complexity of score-based Bayesian network structure learning was extensively studied in the literature [Ordyniak and Szeider, 2013; Korhonen and Parviainen, 2015; Ganian and Korchemna, 2021; Grüttemeier *et al.*, 2021; Grüttemeier and Komusiewicz, 2022].

2 Preliminaries

We use standard terminology for graph theory [Diestel, 2012]. Let $\mathbb{N}$ denote the set of positive integers. We use $[i]$ to denote the set $\{1, \ldots, i\}$. To obtain our results, we will need to consider undirected graphs as well as directed graphs, i.e., digraphs. We use $V(\mathcal{G})$ to denote the vertex set of a (di)graph $\mathcal{G}$. For a directed graph $\mathcal{D}$, we use $A(\mathcal{D})$ to denote its arc set. The *in-neighborhood* of a vertex $X \in V(\mathcal{D})$ is the set $\{\, Y \in V(\mathcal{D}) \mid YX \in A(\mathcal{D}) \,\}$, and the *out-neighborhood* of X is the set $\{\, Y \in V(\mathcal{D}) \mid XY \in A(\mathcal{D}) \,\}$. Elements of these sets are called the *in-* and *out-neighbors* of X and the sizes of these sets are called the *in-degree* and *out-degree* of X, respectively. If $\mathcal{H}$ is a subgraph of a (directed or undirected) graph $\mathcal{G}$, we denote this as $\mathcal{H} \subseteq \mathcal{G}$. The *skeleton* (sometimes called the *underlying undirected graph*) $\underline{\mathcal{D}}$ of a directed graph $\mathcal{D}$ is the simple graph obtained by replacing each arc in $\mathcal{D}$ with an undirected edge. An *edge-cut* in $\underline{\mathcal{D}}$ which separates X from Y is a set $\mathcal{E}$ of edges such that X and Y lie in different connected components of $\underline{\mathcal{D}} - \mathcal{E}$; we say that $\mathcal{E}$ *touches* a vertex Z if it contains at least one edge incident to Z. A *path* between a vertex X_1 and a vertex X_n is a non-repeating sequence of vertices $X_1, \ldots, X_n$ such that for each pair of vertices X_i and X_{i+1}, there is an arc X_iX_{i+1} or $X_{i+1}X_i$ (edge X_iX_{i+1} for an undirected graph). A *causal graph* is a directed acyclic graph $\mathcal{D}$ over a set $V(\mathcal{D})$ of variables (represented as vertices) whose arcs represent causal relationships between these variables. In this paper, we consider variables to range over a finite and fixed domain of size B. A vertex (or variable) X_i is a *collider* on a path $\mathcal{P}$ if the path contains $X_{i-1} \to X_i \leftarrow X_{i+1}$; otherwise it is a *non-collider* [Glymour *et al.*, 2019]. A *directed path* from X_1 to X_n in $\mathcal{D}$ is a path which contains an arc X_iX_{i+1} for each pair X_i and X_{i+1}. If such a path exists, we say that X_n is *reachable* from X_1 or, equivalently, X_n is a *descendant* of X_1 in $\mathcal{D}$. As a special case for $n = 1$, every variable is reachable from (is a descendant of) itself. A *super-structure* $\mathcal{G}$ is a graph on the same vertex set as the causal graph $\mathcal{D}$ which forms a supergraph of the skeleton $\underline{\mathcal{D}}$ of $\mathcal{D}$, i.e., $\underline{\mathcal{D}}$ can be obtained from $\mathcal{G}$ by removing some set of edges. In the context of causal discovery, it is used to represent all potential dependencies between variables (known, e.g., from prior or expert knowledge).

Conditional Independence and d-Separation. Let $\mathcal{S}$ be a subset of variables in $\mathcal{D}$, and let X and Y be two different variables outside of $\mathcal{S}$. We say that X is *d-separated* from Y conditional on $\mathcal{S}$ if and only if all paths between X and Y in $\underline{\mathcal{D}}$ are *blocked* by $\mathcal{S}$. A path $\mathcal{P}$ is *blocked* by $\mathcal{S}$ if at least one of the following two contions holds:

- $\mathcal{P}$ contains a non-collider which is in $\mathcal{S}$, or
- $\mathcal{P}$ contains a collider which has no descendants in $\mathcal{S}$.

Assuming the well-established *Causal Markov Condition* and the *Faithfulness Condition*, X is d-separated from Y by $\mathcal{S}$ if and only if X and Y are conditionally independent given $\mathcal{S}$ [Glymour *et al.*, 2019]. Further, we will denote both of these facts (namely, conditional independence and d-separation of X from Y given $\mathcal{S}$) by $X \perp Y \mid \mathcal{S}$.

The PC algorithm. The PC algorithm is one of the basic algorithms used to determine the Markov Equivalence Class (i.e., a partially oriented skeleton) of the causal graph. It receives as input a set $V(\mathcal{D})$ of variables, possibly a super-structure of the (hidden) causal graph over $V(\mathcal{D})$, and an oracle that can check conditional (in)dependencies between variables X, Y w.r.t. a subset of $V(\mathcal{D}) \setminus \{X, Y\}$.

In its entirety, the PC algorithm consists of two phases: the first phase (called the *learning phase*) computes a skeleton, while the second obtains a partial orientation of the skeleton obtained in the first phase. The key distinction is that while the second phase can be implemented as a polynomial-time postproceessing routine, the learning phase can in general take exponential time. Hence, in this complexity-theoretic study we focus our attention solely to the learning phase, which forms the bottleneck in the worst-case running time of the PC algorithm. This phase consists of the following sequence of procedures [Glymour *et al.*, 2019]:

1. Construct a complete undirected graph over the provided set of variables.
2. Eliminate edges between variables that are either unconditionally independent or are known to be independent due to

expert knowledge (non-adjacent in the super-structure).
3. For each pair of variables $\{X, Y\}$ having an edge between them, and for each variable Z with an edge connected to either of them, eliminate the edge between X and Y if X is conditionally independent of Y given Z.
4. For each pair of variables $\{X, Y\}$ having an edge between them, and for each pair of variables $\{Z, W\}$ with edges both connected to X or both connected to Y, eliminate the edge between X and Y if $X \perp Y \mid \{Z, W\}$.
5. Continue checking independencies conditional on subsets of variables of size $3, \ldots, i$ until there are no more adjacent pairs X, Y, such that there is a subset of variables of size i such that all of the variables in the subset are adjacent to X or all adjacent to Y.

Let $k \leq i$ be the maximum size of sets that need to be considered in Step 5 in order for the PC algorithm to discover the whole skeleton of the causal graph. We remark that if k is known in advance, it can be used to provide a basic upper bound on the running time of the PC algorithm, since we may safely stop the process once all sets of size at most k have been checked.

Various variants of the PC algorithm have been proposed over the last two decades [Glymour *et al.*, 2019]. We will refer to all such variants—and in particular, all complete algorithms that are based on checking the conditional independence of pairs of variables in terms of d-separation in a learning phase, followed by an orientation phase (such as the FCI algorithm [Spirtes *et al.*, 2000])—as *constraint-based* causal discovery algorithms.

Parameterized Complexity. Parameterized complexity theory [Cygan *et al.*, 2015; Downey and Fellows, 2013; Niedermeier, 2006] analyzes the running time of algorithms not only with respect to the input size n, but also to a specified *parameter* $k \in \mathbb{N}$. The high-level idea is to identify parameters which capture the structural properties of instances for computationally challenging problems such that the combinatorial explosion in the running time can be confined to this parameter. The parameterized counterpart to the classical complexity class P is FPT (*fixed-parameter tractable*), which contains all problems that can be solved by an algorithm running in time $f(k) \cdot n^{\mathcal{O}(1)}$, where f is a computable function. Naturally, not all parameterizations give rise to fixed-parameter tractability for problems of interest, and identifying which do is a central task of parameterized analysis.

3 Characterizing Small d-Separating Sets

The aim of this section is to characterize necessary and sufficient conditions for the existence of bounded-size d-separating sets in a hidden causal graph $\mathcal{D}$. While one would at first glance expect that such conditions would be closely tied to structural properties that depend on the orientation of the arcs in $\mathcal{D}$ (and, in fact, some of our results provide precisely such a relationship), here we provide positive results for conditions that are oblivious to the orientation of the arcs, i.e., conditions that solely depend on the undirected skeleton $\underline{\mathcal{D}}$. There are two reasons for this:

1. It is known to be impossible to identify the exact hidden causal graph $\mathcal{D}$ using constraint-based algorithms, but such algorithms can reliably compute $\underline{\mathcal{D}}$ [Eberhardt, 2017, Section 2]. This makes the latter a tangible and guaranteed output of such algorithms.
2. We provide concrete examples and constructions which show that the results linking the existence of bounded-size d-separating sets to undirected structures in $\underline{\mathcal{D}}$ cannot be lifted to analogous directed structures in $\mathcal{D}$.

As a first basic step, it will be useful to provide a lemma that can be seen as an independent short proof of the correctness of the PC algorithm. For two distinct variables X and Y in $\mathcal{D}$, let $\mathcal{D}_{XY}$ and $\underline{\mathcal{D}}_{XY}$ be the subgraphs of $\mathcal{D}$ and $\underline{\mathcal{D}}$ respectively induced on the vertices of the set of all undirected paths between X and Y in $\underline{\mathcal{D}}$. For the following, it will be useful to recall that the causal graphs considered here are acyclic.

Lemma 1. *Let X and Y be distinct and non-adjacent variables such that X is not reachable from Y in $\mathcal{D}$. Let $\mathcal{S}$ be the set of all in-neighbors of Y in $\mathcal{D}_{XY}$. Then $X \perp Y \mid \mathcal{S}$.*

Lemma 1 immediately implies that if the causal graph has in-degree ℓ, then every pair of variables admits a d-separating set of size at most ℓ. In particular, bounded maximum in-degree in $\mathcal{D}$—and hence also bounded maximum degree in $\underline{\mathcal{D}}$—is a sufficient condition for the existence of small d-separating sets. Naturally, such a condition is far from necessary: for instance, the class of simple stars with all edges oriented towards the center has arbitrarily large maximum in-degree even though each pair of non-adjacent variables admits a d-separating set of size 0.

As our next task, we will provide a stronger sufficient condition on $\underline{\mathcal{D}}$ for the existence of small d-separating sets—specifically, the existence of a bounded number of edge-disjoint paths between each pair of variables. The next lemma provides the first step towards this goal:

Lemma 2. *Let X and Y be non-adjacent vertices in $\underline{\mathcal{D}}$ such that X is not reachable from Y in $\mathcal{D}$. Assume that $\underline{\mathcal{D}}$ admits an edge-cut of size k which separates X and Y and does not touch X. Then X and Y can be d-separated by a set of size at most $2k$ in $\mathcal{D}$.*

We note that it is crucial that the edge-cut considered in Lemma 2 does not touch X: Figure 1 depicts a causal graph admitting an edge-cut $\{XZ, XW\}$ of size two, but requires large d-separating set for X and Y.

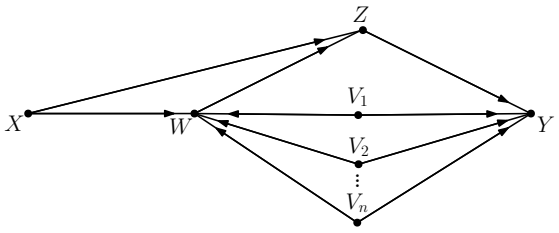

Figure 1: An example of a causal graph where X and Y are separated by two edges XZ and XW, but any d-separating set for X and Y has size at least $n + 1$.

Interestingly, when aiming for a sufficient condition that would guarantee the existence of small d-separating sets between *all* pairs of variables, we show that the existence of arbitrary

edge-cuts of bounded size is sufficient (regardless of which variables they touch).

Theorem 3. *If every pair of vertices X and Y can be separated in $\underline{\mathcal{D}}$ by an edge-cut of size at most k, then every pair of non-adjacent variables can be d-separated in $\mathcal{D}$ by a set of size at most $2k^2$.*

Notice that the bound above is imposed on the number of undirected edge-disjoint paths. By contrast, restricting the number of directed edge-disjoint paths does not suffice: as a simple example, consider a causal graph $\mathcal{L}_n$ that consists of two variables X and Y plus n vertex-disjoint paths of length 2 between them, where every vertex other than X and Y is a source. In particular, X and Y are sinks.

Observation 4. *$\mathcal{L}_n$ does not contain a directed path between any pair of non-adjacent variables, but the only d-separating set for X and Y has size n.*

However, this still raises the question of whether Theorem 3 could be strengthened by simply requiring the existence of a vertex-separator of size at most k (instead of having an edge-cut of size at most k) between each pair of variables. Below, we answer this in the negative:

Theorem 5. *There exists a family $\{\mathcal{D}_i : i \in \mathbb{N}\}$ of causal graphs with $|V(\mathcal{D}_i)| = \mathcal{O}(4^i)$, $i \in \mathbb{N}$, such that each $\mathcal{D}_i$ contains a pair of non-adjacent variables with no d-separating set of size smaller than $2^{i-1}+1$, and yet each pair of variables in $\mathcal{D}_i$ admits a vertex separator of size 3.*

Proof Sketch. For convenience, let us define $\mathcal{D}_0$ to simply be a causal graph consisting of a single arc. If the edge begins in X and ends in Y, we say that $\mathcal{D}_0$ has *orientation* XY. We construct the family by induction; to this end, let us assume that the causal graph $\mathcal{D}_i$, $i \in \mathbb{N} \cup \{0\}$, has already been defined and has a special pair of vertices called its orientation.

For the construction itself, take two vertex-disjoint copies of $\mathcal{D}_i$ with orientations XR and $R'Y$ and glue them together by identifying R and R', and denote the obtained graph as $\mathcal{D}_i^{\text{repeat}}$. Take another two vertex-disjoint copies of $\mathcal{D}_i$ with orientations $M'X'$ and MY' and glue them together by identifying M and M', and denote the graph as $\mathcal{D}_i^{\text{mirror}}$. Finally, glue together $\mathcal{D}_i^{\text{repeat}}$ and $\mathcal{D}_i^{\text{mirror}}$ by identifying X' with X and Y' with Y. The resulting graph is $\mathcal{D}_{i+1}$ with orientation XY. An illustration of the construction—which may be viewed as "fractal" in nature—is provided in Figure 2.

To complete the proof, it now remains to establish the following two claims:

Claim 1. For every $i \in \mathbb{N}$, $\mathcal{D}_i$ admits a vertex separator of size at most 3 between any pair of vertices.

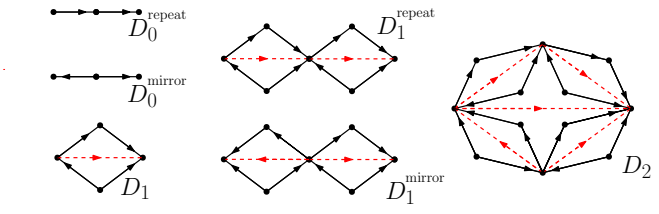

Figure 2: Construction of $\mathcal{D}_{i+1}$ from 4 copies of $\mathcal{D}_i$ for $i = 0, 1$. The red dashed arcs depict the orientations.

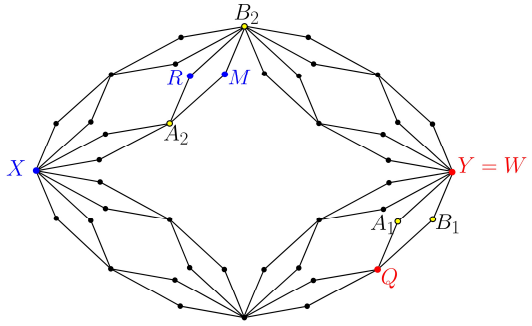

Figure 3: Vertex separators in the skeleton of $\mathcal{D}_3$ from the proof of Claim 1. Red vertices form the vertex separator between A_1 and B_1, while blue vertices form the vertex separator between A_2 and B_2.

Claim 2. For every $i \in \mathbb{N}$, the causal graph $\mathcal{D}_i$ with orientation XY contains no d-separating set between X and Y of size smaller than $2^{i-1}+1$.

The proof of Claim 1 is based on case analysis for all pairs of variables, and is illustrated on Figure 3. The key insight underlying the proof of Claim 2 is that at each step from $\mathcal{D}_i$ to $\mathcal{D}_{i+1}$, the size of a minimum d-separating set can be shown to increase by a factor of 2. □

Theorems 3 and 5 imply that while having a bounded number of undirected edge-disjoint paths between variables guarantees the presence of a bounded-size d-separating set, this is not the case for undirected vertex-disjoint paths. Hence, having a bounded number of undirected edge-disjoint paths can be seen as a *sufficient* condition for the existence of a bounded-size d-separating set. Our aim in the rest of this section is to identify a condition which is *necessary*.

We begin by making a simple observation concerning a necessary condition tied to the hidden causal graph (as opposed to the skeleton). In particular, since a d-separating set must intersect each directed path between a pair of variables in the causal graph, we obtain:

Observation 6. *If a causal graph $\mathcal{D}$ admits a d-separating set of size k for each pair of non-adjacent variables, then for each pair of non-adjacent variables the number of directed vertex-disjoint paths is bounded by the same bound k.*

In terms of conditions on the actual skeleton $\underline{\mathcal{D}}$, we note the following. If the number of vertex-disjoint paths between two non-adjacent variables X and Y in $\underline{\mathcal{D}}$ is large, then there exists an acyclic orientation of $\underline{\mathcal{D}}$ with no small d-separating set between X and Y (this occurs, e.g., when all of the X-Y paths are oriented from X to Y). Hence, having a small number of vertex-disjoint paths between each pair of non-adjacent variables is necessary to guarantee that every orientation of $\underline{\mathcal{D}}$ admits a small d-separating set. In other words:

Corollary 7. *Let $\underline{\mathcal{D}}$ be an undirected graph. Assume that, in every acyclic orientation $\mathcal{D}$ of $\underline{\mathcal{D}}$, every pair of non-adjacent variables can be d-separated in $\mathcal{D}$ by a set of size k. Then every pair of non-adjacent variables can be separated in $\underline{\mathcal{D}}$ by a vertex separator of size at most k.*

Finally, the fact that Corollary 7 references *every* orientation $\mathcal{D}$ of $\underline{\mathcal{D}}$ is crucial. Indeed:

Lemma 8. *There exists a family of undirected graphs $\mathcal{H} = \{ \underline{\mathcal{X}_i} \mid i \in \mathbb{N} \}$ with the following properties:* (I) *Each $\underline{\mathcal{X}_i}$ admits an acyclic orientation $\mathcal{X}_i$ where every pair of non-adjacent variables can be d-separated by a set of size 0, and* (II) *$\underline{\mathcal{X}_i}$ contains a pair of non-adjacent variables A, B whose minimum vertex separator has size i.*

As the final result in this section, we also exclude the possibility of generalizing Observation 6 to a bound on the number of directed edge-disjoint paths (as opposed to a bound on the number of directed vertex-disjoint paths).

Lemma 9. *There exists a family $\{\mathcal{D}'_i : i \in \mathbb{N}\}$ of causal graphs with $|V(\mathcal{D}'_i)| = \mathcal{O}(4^i)$, $i \in \mathbb{N}$, such that each $\mathcal{D}_i$ contains 2^i directed edge-disjoint paths between some pair of non-adjacent variables, and yet each pair of non-adjacent variables can be d-separated by a set of size at most 3.*

Proof Sketch. For every $i \in \mathbb{N}_0$, we define the causal graph $\mathcal{D}'_i$ as follows. The skeleton of $\mathcal{D}'_i$ coincides with the skeleton of $\mathcal{D}_i$ from Theorem 5, and if such $\mathcal{D}_i$ has orientation XY, in $\mathcal{D}'_i$ we orient all the arcs on every path from X and Y towards Y. In particular, $\mathcal{D}'_{i+1}$ can be obtained by gluing together 4 copies of $\mathcal{D}'_i$, $i \in \mathbb{N}_0$. To complete the proof, it suffices to show that there are constant-sized d-separating sets between each pair of variables in $\mathcal{D}'_i$; at that point, the lemma follows from the fact that each causal graph $\mathcal{D}'_i$ consists of 2^i edge-disjoint directed paths from X to Y. □

4 Limits of Tractability for the PC Algorithm

While the previous section was aimed at identifying necessary and sufficient conditions for the existence of a small d-separating set, having a small d-separating set itself is merely a *necessary* condition for efficient causal discovery. Indeed, in this section we show that the existence of a small d-separating set cannot on its own guarantee efficient running times for causal discovery.

We begin with a lower bound ruling out fixed-parameter causal discovery by any constraint-based algorithm:

Lemma 10. *There exists a class of causal graphs $\mathcal{H} = \{ \mathcal{D}_{i,k} \mid i, k \in \mathbb{N}; k \leq i \}$ such that:* (I) *each $\mathcal{D}_{i,k}$ contains $i + 3$ vertices,* (II) *each $\mathcal{D}_{i,k}$ has a single pair X, Y of non-adjacent vertices with a d-separating set of size k, and* (III) *on each $\mathcal{D}_{i,k}$, no constraint-based algorithm can determine whether X and Y are conditionally independent in time less than $\Theta(n^{k-1})$.*

While Lemma 10 may already seem disheartening, the lower bound it provides only becomes truly problematic as k grows above a small constant. In other words, at this point it would still seem plausible that the PC algorithm achieves a running time of, e.g., $n^{\mathcal{O}(k)}$, where k is the size of the largest d-separating set in the causal graph. As our second result in this section, we show that the situation is in fact much worse—there is a class of causal graphs for which the PC algorithm will necessarily run in exponential time, even though all pairs of variables admit very small d-separating sets between them.

Theorem 11. *For each $n \geq 2$, there exists a causal graph $\mathcal{T}_n$ on $\mathcal{O}(2^n)$ vertices such that $\mathcal{T}_n$ admits d-separating sets of size at most 3 between every pair of non-adjacent variables, but the PC algorithm only discovers a d-separating set of size $2^n + 2$ for at least one pair of variables.*

Proof Sketch. To obtain $\mathcal{T}_n$, we start from two full binary trees $\mathcal{T}_n^1$ and $\mathcal{T}_n^2$ with 2^n leaves each, where all the arcs are oriented from the roots to the leaves. We glue $\mathcal{T}_n^1$ and $\mathcal{T}_n^2$ together at their roots and denote the resulting common root by O_n. Moreover, we introduce a variable W_n and add an incoming arc to it from every leaf of $\mathcal{T}_n^1$. Symetrically, we introduce a variable E_n and add an incoming arc to it from every leaf of $\mathcal{T}_n^2$. Finally, we add two variables N_n and S_n along with out-going arcs to X and Y from both of them, see Figure 4 for an illustration.

Now, let us analyze how the PC algorithm will discover the d-separating sets in $\mathcal{T}_n$. In the first stage, it searches for d-separating sets of size 0, which will disconnect S_n and N_n from all the other vertices except E_n and W_n. It will then proceed by searching for d-separating sets in the neighborhoods of each pair of variables which still have an edge between them, and one can show that this results in the discovery of $\mathcal{T}_n$ plus one additional edge W_nE_n by the time it completes its check for d-separating sets of size at most 3. However, to eliminate this final edge, the PC algorithm will need to use a d-separating set containing N_n, S_n and all the leaves of $\mathcal{T}_n^1$ or $\mathcal{T}_n^2$. Such a set has size at least $2^n + 2$. □

5 Utilizing the Super-Structure for Fixed-Parameter Algorithms

In our final section, we show that the lower bounds arising from Lemma 10 and Theorem 11 can both be circumvented if we are provided with a super-structure possessing suitable structural properties. In particular, our aim is to show that causal discovery can be carried out in fixed-parameter time not with respect to the size of a maximum d-separating set (as this was ruled out in Lemma 10), but instead with respect to a structural measure of a provided super-structure.

A natural graph-theoretic measure of the super-structure is the well-established *treewidth*—a fundamental graph parameter which intuitively captures the "tree-likeness" of a graph [Robertson and Seymour, 1986]. However, we can immediately rule out the use of treewidth (and also many other related graph parameters) by recalling the graph class

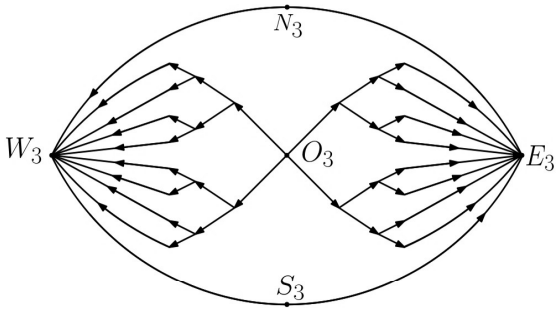

Figure 4: The causal graph $\mathcal{T}_3$, where E_3 and W_3 admit a d-separating set of size 3 but the PC algorithm only discovers a d-separating set of size 10.

$\mathcal{L} = \{ \mathcal{L}_i \mid i \in \mathbb{N} \}$ used in Observation 4. Indeed, X and Y form a vertex cover of each graph $\mathcal{L}_i \in \mathcal{L}$, and hence all graphs in that class have treewidth at most 2. However, the only d-separating set between X and Y has size i. Thus, by setting $\underline{\mathcal{L}}'_i$ to be a super-structure for $\mathcal{L}_i$ that contains each edge in $\underline{\mathcal{L}}_i$ plus the edge XY, we obtain the corollary below. In particular, each $\mathcal{L}_i$ consists of X, Y, and a set of i sources adjacent to both X and Y, and $\underline{\mathcal{L}}'_i$ adds a "superfluous" edge XY to the skeleton of $\mathcal{L}_i$.

Corollary 12. *For every $i \in \mathbb{N}$, there exists a super-structure $\underline{\mathcal{L}}'_i$ of $\mathcal{L}_i$ with a vertex cover of size 2 such that no constraint-based causal discovery algorithm can compute the skeleton of the hidden causal graph $\mathcal{L}_i$ in time less than B^i.*

We remark that the same construction rules out efficient algorithms based on not only the treewidth and the vertex cover number, but also pathwidth, treedepth [Nesetril and de Mendez, 2012], tree-cut width [Ganian *et al.*, 2022] and a range of other graph parameters typically used to achieve fixed-parameter tractability. However, as we will see in the rest of the section, there are natural structural properties of the super-structure which guarantee efficient causal discovery. The first such property is the maximum degree in the super-structure. It may be worth noting that it is exceedingly rare for a computational problem to be fixed-parameter tractable w.r.t. the maximum degree and yet intractable w.r.t. parameters such as the vertex cover number.

Theorem 13. *If the PC algorithm is provided an n-variable super-structure $\mathcal{G}$ of maximum degree k, then it will discover the skeleton $\underline{\mathcal{D}}$ of the hidden causal graph in time $2^k B^k \mathcal{O}(nk)$, where B is the size of domain.*

Next, we show that the tractability result of Theorem 13 can be extended to the setting where only the maximum degree of each 2-connected component is bounded. To this end, we define a new constraint-based algorithm PC*. For a super-structure $\mathcal{G}$ and two variables $X, Y \in V(\mathcal{G})$, let $\mathcal{G}_{XY}$ be the subgraph of $\mathcal{G}$ induced by all paths between X and Y in $\mathcal{G}$. PC* takes $\mathcal{G}$ (which may, in the worst case, be the complete graph) as input and proceeds similarly to the PC algorithm, with only a single modification: while looking for potential d-separating sets for X and Y, it only checks those subsets of neighbors of Y that belong to $\mathcal{G}_{XY}$.

Theorem 14. *Given a super-structure $\mathcal{G}$ of $\mathcal{D}$ over n variables, PC* computes the skeleton of $\mathcal{D}$ in time $2^k B^k \mathcal{O}(nk)$, where k is the maximum degree over all 2-connected components of $\mathcal{G}$.*

Even though the difference between the PC and PC* algorithms might seem rather subtle, the latter immediately extends the class of instances for which the skeleton can be computed efficiently. This can be illustrated, e.g., on the class $\{ \mathcal{L}^n_k \mid n, k \in \mathbb{N} \}$ of causal graphs obtained from the graphs $\mathcal{L}_k$ (cf. Observation 4) by introducing $n - k$ new in-neighbors of X and $n - k$ new in-neighbors of Y.

Corollary 15. *If $\mathcal{L}^n_k$ is a hidden causal graph and $\mathcal{G}$ contains the only extra edge between its pair of high-degree vertices, then the* PC *algorithm requires $\Omega(n^{k-1})$ iterations to eliminate the single superfluous edge in $\mathcal{G}$, while the* PC* *algorithm computes the skeleton in time $2^k B^k \mathcal{O}(nk)$.*

As our final contribution, we provide a new causal discovery algorithm called PC$_{\text{cut}}$ that features fixed-parameter runtime guarantees not only on all instances where the PC and PC* algorithms do, but also on more general classes of inputs. To this end, it will be useful to recall that a bound on the size of smallest edge-cuts between each pair of variables is a sufficient condition for the existence of small d-separating sets (see Theorem 3). While causal graphs with this property need not be efficiently discoverable by constraint-based algorithms, this changes when dealing with a provided super-structure.

The PC$_{\text{cut}}$ algorithm alters the original PC algorithm by testing conditional independence via subsets selected from a smallest edge-cut between the variables X, Y, instead of simply checking the neighborhood of these two variables. More precisely, we define the PC$_{\text{cut}}$ algorithm as follows. It begins by performing the same initial two steps as the PC algorithm (see Section 2). Afterwards, it performs a subroutine that starts with $j := 1$ and increments this up to a value of $2k^2$, where k upper-bounds the size of a smallest edge-cut. For each choice of j, it runs over all ordered pairs of adjacent variables, and for each such pair (X, Y) it computes an edge-cut $\mathcal{E}$ of minimum size between X and Y in the graph computed so far (but without the edge XY) that does not touch X (this can be done, e.g., via the Ford-Fulkerson algorithm where we make the edges incident to X undeletable). Let $\mathcal{S}^{\mathcal{E}}$ be the set of variables incident to the edges in $\mathcal{E}$. We now perform essentially the same procedure the PC algorithm originally performs over the neighborhood of X and Y, but on the set $\mathcal{S}^{\mathcal{E}}$; in particular, for all subsets $\mathcal{S}$ of $\mathcal{S}^{\mathcal{E}}$ of size j, we test whether $X \perp Y \mid \mathcal{S}$ and if this test succeeds we eliminate the edge XY.

Theorem 16. *When given a super-structure $\mathcal{G}$ of an n-variable causal graph $\mathcal{D}$ such that each pair of variables can be separated by an edge-cut of size at most k in $\mathcal{G}$, the* PC$_{\text{cut}}$ *algorithm will compute the skeleton $\underline{\mathcal{D}}$ of $\mathcal{D}$ in time at most $2^{2k^2} B^{2k^2} \mathcal{O}(nk)$.*

We conclude this section by noting that there exist classes of inputs where PC$_{\text{cut}}$ will outperform both the PC and PC* algorithms by an arbitrarily large factor. For instance, this occurs on the class of causal graphs depicted in Figure 4.

Corollary 17. *If $\mathcal{T}_n$ is a hidden causal graph and $\mathcal{G}$ contains the only extra edge between its pair of high-degree vertices,* PC *and* PC* *need $2^{\Omega(2^n)}$ iterations to eliminate the single extra edge, while* PC$_{\text{cut}}$ *computes the skeleton in time $B^{\mathcal{O}(1)} n$.*

6 Concluding Remarks

Our results yield deep insights into how causal graph structures impact constraint-based causal discovery algorithms. We have provided detailed near-tight characterizations of causal graphs allowing small d-separating sets, identified obstacles for the PC algorithm to run efficiently, and established fixed-parameter tractability by exploiting properties of the super-structure. However, it remains open whether a skeleton with many edge-disjoint paths between variable pairs implies the existence of a large d-separating set orientation. Although the immediate practical implications of our results may be limited, we believe they will inspire novel approaches for the foundational as well as empirical aspects of causal discovery.

Acknowledgments

The authors are grateful for and acknowledge support from the Austrian Science Fund (FWF, Projects 10.55776/Y1329, 10.55776/P36688, 10.55776/P36420, and 10.55776/COE12) as well as from the Vienna Science and Technology Fund (WWTF, Project ICT22-029).

References

[Chen *et al.*, 2016] Eunice Yuh-Jie Chen, Yujia Shen, Arthur Choi, and Adnan Darwiche. Learning Bayesian networks with ancestral constraints. *Advances in Neural Information Processing Systems*, 29, 2016.

[Claassen *et al.*, 2013] Tom Claassen, Joris M. Mooij, and Tom Heskes. Learning sparse causal models is not NP-hard. *CoRR*, abs/1309.6824, 2013.

[Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[Diestel, 2012] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[Downey and Fellows, 2013] Rodney G. Downey and Michael R. Fellows. *Fundamentals of parameterized complexity*. Texts in Computer Science. Springer Verlag, 2013.

[Eberhardt, 2017] Frederick Eberhardt. Introduction to the foundations of causal discovery. *Int. J. Data Sci. Anal.*, 3(2):81–91, 2017.

[Ganian and Korchemna, 2021] Robert Ganian and Viktoriia Korchemna. The complexity of Bayesian network learning: Revisiting the superstructure. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 430–442, 2021.

[Ganian *et al.*, 2022] Robert Ganian, Eun Jung Kim, and Stefan Szeider. Algorithmic applications of tree-cut width. *SIAM J. Discret. Math.*, 36(4):2635–2666, 2022.

[Glymour *et al.*, 2019] Clarc Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10:Sec. Statistical Genetics and Methodology, 2019.

[Grüttemeier and Komusiewicz, 2022] Niels Grüttemeier and Christian Komusiewicz. Learning Bayesian networks under sparsity constraints: A parameterized complexity analysis. *J. Artif. Intell. Res.*, 74:1225–1267, 2022.

[Grüttemeier *et al.*, 2021] Niels Grüttemeier, Christian Komusiewicz, and Nils Morawietz. On the parameterized complexity of polytree learning. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4221–4227. ijcai.org, 2021.

[Korhonen and Parviainen, 2015] Janne H Korhonen and Pekka Parviainen. Tractable Bayesian network structure learning with bounded vertex cover number. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[Mokhtarian *et al.*, 2021] Ehsan Mokhtarian, Sina Akbari, AmirEmad Ghassami, and Negar Kiyavash. A recursive Markov boundary-based approach to causal structure learning. In Thuc Duy Le, Jiuyong Li, Gregory Cooper, Sofia Triantafyllou, Elias Bareinboim, Huan Liu, and Negar Kiyavash, editors, *The KDD 2021 Workshop on Causal Discovery, CD@KDD 2021, Singapore, August 15, 2021*, volume 150 of *Proceedings of Machine Learning Research*, pages 26–54. PMLR, 2021.

[Mokhtarian *et al.*, 2022] Ehsan Mokhtarian, Sina Akbari, Fateme Jamshidi, Jalal Etesami, and Negar Kiyavash. Learning Bayesian networks in the presence of structural side information. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 7814–7822. AAAI Press, 2022.

[Nesetril and de Mendez, 2012] Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and Combinatorics*. Springer, 2012.

[Ng *et al.*, 2021] Ignavier Ng, Yujia Zheng, Jiji Zhang, and Kun Zhang. Reliable causal discovery with improved exact search and weaker assumptions. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20308–20320, 2021.

[Niedermeier, 2006] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford, 2006.

[Ordyniak and Szeider, 2013] Sebastian Ordyniak and Stefan Szeider. Parameterized complexity results for exact Bayesian network structure learning. *J. Artif. Intell. Res.*, 46:263–302, 2013.

[Perrier *et al.*, 2008] Eric Perrier, Seiya Imoto, and Satoru Miyano. Finding optimal Bayesian network given a superstructure. *J. Mach. Learn. Res.*, 9:2251–2286, 2008.

[Ramaswamy and Szeider, 2022] Vaidyanathan Peruvemba Ramaswamy and Stefan Szeider. Learning large Bayesian networks with expert constraints. In James Cussens and Kun Zhang, editors, *38th Conference on Uncertainty in Artificial Intelligence (UAI 2022), Eindhoven, Netherlands, August 1–5, 2022*, Proceedings of Machine Learning Research (PMLR), page 180:1592–1601, 2022.

[Robertson and Seymour, 1986] Neil Robertson and P. D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

[Shalizi, 2023] Cosma Rohilla Shalizi. *Advanced Data Analysis from an Elementary Point of View*. Carnegie Mellon University, 2023. https://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/.

[Spirtes *et al.*, 2000] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search, Second Edition*. Adaptive computation and machine learning. MIT Press, 2000.

[Talvitie and Parviainen, 2020] Topi Talvitie and Pekka Parviainen. Learning Bayesian networks with cops and robbers. In Manfred Jaeger and Thomas Dyhre Nielsen, editors, *International Conference on Probabilistic Graphical Models, PGM 2020, 23-25 September 2020, Aalborg, Hotel Comwell Rebild Bakker, Skørping, Denmark*, volume 138 of *Proceedings of Machine Learning Research*, pages 473–484. PMLR, 2020.

[Tian *et al.*, 1998] Jing-Jing Tian, Azaria Paz, and Judea Pearl. Finding minimal D-separators. Technical Report R-254, UCLA, February 1998.

[Tsamardinos *et al.*, 2006] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

[Wienöbst and Liskiewicz, 2021] Marcel Wienöbst and Maciej Liskiewicz. An approach to reduce the number of conditional independence tests in the PC algorithm. In Stefan Edelkamp, Ralf Möller, and Elmar Rueckert, editors, *KI 2021: Advances in Artificial Intelligence - 44th German Conference on AI, Virtual Event, September 27 - October 1, 2021, Proceedings*, volume 12873 of *Lecture Notes in Computer Science*, pages 276–288. Springer, 2021.