Finding minimal d-separators in linear time and applications*

Benito van der Zander
Theoretical Computer Science
University of Lübeck, Germany
benito@tcs.uni-luebeck.de

Maciej Liśkiewicz
Theoretical Computer Science
University of Lübeck, Germany
liskiewi@tcs.uni-luebeck.de

Abstract

The study of graphical causal models is fundamentally the study of separations and conditional independences. We provide linear-time algorithms for two graphical primitives: to test, if a given set is a minimal d-separator, and to find a minimal d-separator in directed acyclic graphs (DAGs), completed partially directed acyclic graphs (CPDAGs) and restricted chain graphs (RCGs) as well as minimal m-separators in ancestral graphs (AGs). These algorithms improve the runtime of the best previously known algorithms for minimal separators that are based on moralization and thus require quadratic time to construct and handle the moral graph. (Minimal) separating sets have important applications like finding (minimal) covariate adjustment sets or conditional instrumental variables.

1 INTRODUCTION

One of the key factors stimulating the recent progress in causal research was the development of graphical causal models, like e.g., those based on directed acyclic graphs (DAGs), which allow intuitive, mathematically sound modeling of causal relationships. Likewise, the models and the related graphical notions used in modern causality admit to apply discrete- and graph-algorithmic techniques developed in computer science to provide algorithmic support for causal analysis. Perhaps, the most prominent example here is the concept of *d-separation* [12] – the notion of central importance in graphical causal models and Bayesian networks that is deeply connected with the probabilistic concept of conditional independence [28] and simultaneously allows to use algorithmic frameworks to analyze conditional independences in causal models. Algorithmic aspects to find and verify d-separators and its generalizations are the main subject of this paper.

Recall, that graphical separation is defined as follows. A path π in a DAG is said to be blocked by a set of nodes $\mathbf{Z}$ if and only if π contains a chain $A \to B \to C$ or a fork $A \leftarrow B \to C$ such that the middle node B is in $\mathbf{Z}$, or π contains a collider $A \to B \leftarrow C$ such that the middle node B is not in $\mathbf{Z}$ and such that no descendant of B is in $\mathbf{Z}$. A set $\mathbf{Z}$ is said to d-separate a node X from Y if and only if $\mathbf{Z}$ blocks every path between X and Y.

undirected path π ↕

1.1 Reductions to d-separations

D-separation plays a major role in many aspects of causal research. In particular, d-separation allows to specify the set of probability distributions which are compatible with a DAG. This involves perhaps the most direct application of graphical separation – *model testing* with the aim to check whether a graphical causal model is in fact consistent with the probability distribution P it is intended to represent. This is a necessary step before any conclusions can be drawn from the model, because if the model is not consistent to the data, most conclusions will be worthless. In practice, such a test verifies conditional independence for every non-adjacent pair of nodes X and Y given a canonical basis d-separation set $\mathbf{Z}$ [6]. For instance, Pearl and Meshkat [14] discuss for hierarchical linear regression models *parental* basis sets, i.e., those containing all parents of Y. They also consider sets that contain only nodes whose distance to Y is smaller than the distance between X and Y. This leads to obtaining separators which can be substantially smaller than the parental basis sets and thus can decrease the amount of independences that need to be tested. In general, a challenging task of model testing is to detect for any given

* This work was supported by the Deutsche Forschungsgemeinschaft (DFG) grant LI 634/4-2.

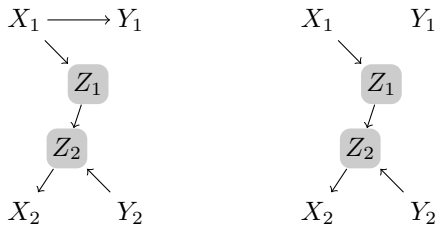

Figure 1: A reduction of testing if $\mathbf{Z} = \{Z_1, Z_2\}$ is a valid and minimal covariate adjustment relative to $\mathbf{X} = \{X_1, X_2\}$ and $\mathbf{Y} = \{Y_1, Y_2\}$ in a DAG (left) to testing if $\mathbf{Z}$ is a minimal d-separator between $\mathbf{X}$ and $\mathbf{Y}$ in the so called *proper back-door graph* [25].

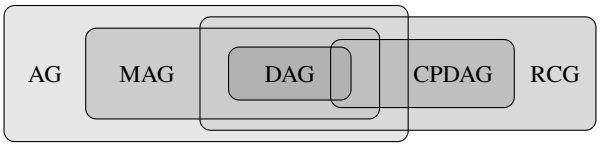

Figure 2: The inclusion relationships between the relevant classes of causal graphs for which our algorithms test and find minimal separators in time $\mathcal{O}(n+m)$, where n denotes the number of nodes and m the number of edges in a causal structure. We have proper inclusions: DAG $\subsetneq$ RCG, CPDAG $\subsetneq$ RCG, DAG $\subsetneq$ MAG $\subsetneq$ AG. This means, e.g., that every DAG is an RCG but not necessarily a CPDAG.

pair of nodes a *minimal* or *minimum* d-separator.

Another fundamental problem of causal analysis, which can be reduced to computing d-separators, is estimation of causal effects from non-experimental data via *covariate adjustment sets*. The causal effect $P(\mathbf{y}|do(\mathbf{x}))$ describes how some outcome variables $\mathbf{Y}$ will change, if some exposure variables $\mathbf{X}$ are changed. Given a causal graph it is often possible to estimate the causal effect from the observed probability distribution $P(\mathbf{v})$. When the causal effect can be calculated using adjustment, $P(\mathbf{y}|do(\mathbf{x})) = \sum_{\mathbf{z}} P(\mathbf{y}|\mathbf{x}, \mathbf{z})P(\mathbf{z})$, the set $\mathbf{Z}$ is called an adjustment set. Though this method of identification is not complete – do-calculus [13] can permit identification even if covariate adjustment is impossible – it provides an effective option for effect estimation due to its well understood statistical properties. Recently we have proved [24, 25] that causal effect identification by covariate adjustment with multiple exposures and outcomes and in the presence of latent confounding can be reduced efficiently to the problem of d-separation in a DAG. In particular, those results yield practically implementable algorithms, which in linear time, with respect to the size of a DAG, reduce testing and finding of adjustment sets to testing and finding of d-separators in a DAG. For an example illustrating such reduction, see Fig. 1.

Moreover, we [24, 25] show that the reduction can be generalized to m-separators in maximal ancestral graphs (MAGs) – the model specifying the ancestral relations between the variables but allowing that not all variables are included in the model [18]. The subsequent research has resulted in further extensions of this reduction technique for: completed partially directed acyclic graphs (CPDAGs) [17, 23], partial ancestral graphs (PAGs) [17], chain graphs (CGs) and restricted chain graphs (RCGs) [23], and maximally oriented partially directed acyclic graphs (PDAGs) [16]. CPDAGs [1] (resp. PAGs) represent classes of causal models where a single graph represents all Markov equivalent DAGs (MAGs), i.e., those having the same conditional independences. PDAGs are CPDAGs with background knowledge [11] and they are known in the literature under various names such as, e.g. interventional essential graphs [7]. CGs [10] generalize CPDAGs by representing some Markov equivalent DAGs, and RCGs constitute a broad subclass of CGs [23]. For the inclusion relationships between some of these models, see Fig. 2.

As a last example of a canonical problem that can be reduced to d-separations in DAGs, we discuss testing and finding of conditional instrumental variables and generalized instrumental sets. The classical instrumental variables (IVs) are a widely used approach to identify parameters in linear models [3, 2, 9]. This is a sufficient but not necessary tool for parameter identification: IVs are often not applicable even if the parameters are identifiable. In [13] Pearl gave a generalization of the method to *conditional* IVs and Brito [4] and Brito and Pearl [5] have extended the IVs to *instrumental sets* that allow the identification of multiple parameters simultaneously. However, one of the barriers to the application of these generalized IV based tools is of algorithmic nature. First steps towards implementability of these methods have been shown in [26] and [22], where in both cases the problem to find a conditional IV, resp. an instrumental set, have been reduced to testing and finding of a so called *nearest separator* – a d-separating set which satisfies some additional desirable constraints. In [26] we show that for a given DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ of n nodes $\mathbf{V}$ and m edges $\mathbf{E}$, a nearest separator can be computed in time $\mathcal{O}(nm)$ (if such a separator exists). As a consequence, this implies that a conditional IV can be found in time $\mathcal{O}(n^2m)$ and simple instrumental sets in time $\mathcal{O}(nm)$. Finding a general instrumental set seems to be harder: In [22] we show that testing if a given set $\mathbf{Z}$ is a generalized instrumental set is NP-complete.

1.2 Our contribution

It is well known that in a given DAG $\mathcal{G}$ one can find a d-separator relative to X and Y in time $\mathcal{O}(n+m)$. For example, the set containing all ancestors of X and Y (except themselves) is such a separator (see, e.g., [25]). On the other hand finding separator sets which satisfy some desirable constraints, like minimality, becomes a much more challenging task. Recall, that $\mathbf{Z}$ is minimal if no proper subset of $\mathbf{Z}$ d-separates X and Y in $\mathcal{G}$.

First nontrivial algorithms for testing and for finding a minimal d-separator in a DAG were proposed by Tian, Paz, and Pearl more than two decades ago [21]. To solve the problems, both algorithms compute first a moral graph (for a definition see Sec. 2) of a subgraph of $\mathcal{G}$ induced by all ancestors of X and Y, and then reduce the original problems to testing, resp. to finding minimal vertex separators in undirected graphs. The total time complexity of this approach is $\mathcal{O}(m_a)$, where m_a denotes the number of edges in the moral graph.

The algorithms of Tian et al. have remained the most efficient known ones so far. Unfortunately, for some DAGs of n nodes and m edges, the number of edges m_a in the moral graph can reach a value proportional to m^2. For example, the DAG $\mathcal{G}$ over nodes $\{X, Y, V_1, \ldots, V_n\}$ and with edges $X \to V_i$ and $V_i \to Y$, for $i = 1, \ldots, n$, has $m = 2n$ edges, while the corresponding moral graph has $m_a = n(n+1)/2 + 2n$ undirected edges.

In this paper we show, to the best of our knowledge for the first time, how to test and find minimal d-separators in DAGs directly, i.e., without going through moral graphs and such that the running time of the algorithms is linear in the number of edges in $\mathcal{G}_{An(\mathbf{X}\cup\mathbf{Y})}$ – the subgraph of $\mathcal{G}$ induced by all ancestors of X and Y. We address general variants of these problems requiring $\mathbf{X}$ and $\mathbf{Y}$ to be (non-empty) disjoint subsets of $\mathbf{V}$ and $\mathbf{Z}$ to be minimal as well as imposing the constraint $\mathbf{I} \subseteq \mathbf{Z} \subseteq \mathbf{R}$ for given subsets $\mathbf{I}, \mathbf{R}$ of $\mathbf{V}$, with $\mathbf{I} \subseteq \mathbf{R}$. Thus, the nodes in $\mathbf{I}$ (possible empty) are required to be always included in the d-separating set, even if the set remains a separator without these nodes. The set $\mathbf{V} \setminus \mathbf{R}$ corresponds to a set of unobserved variables. Furthermore, we show how to extend our approach to test and find in linear time separators in more general causal structures: AGs, MAGs, CPDAGs, and restricted chain graphs.

As a direct consequence, our results improve those of [17, 23, 24, 25] reducing, from quadratic to linear, the running times of algorithms to test and find minimal adjustment sets for identification of causal effects with multiple exposures and outcomes and in the presence of latent confounding in DAGs, MAGs, CPDAGs, and restricted chain graphs. Moreover, they allow to decrease by a factor up to n the running time to test and to find conditional instrumental variables as well as instrumental sets, which improves the result of [26] and [22].

The paper is organized as follows: Section 2 introduces notation and standard concepts of graphical models. Section 3 explains informally how our algorithms work on DAGs. Section 4 proves properties of minimal separators, which are used in Section 5 to show the correctness of the algorithms and to generalize them to AGs and RCGs. Section 6 continues the applications mentioned in Subsection 1.1. Section 7 concludes the paper. The omitted proofs can be found in the supplementary material.

2 PRELIMINARIES

We consider mixed graphs $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ with $n = |\mathbf{V}|$ nodes (variables) and $m = |\mathbf{E}| \geq n$ edges of the form $\to$, $—$, or $\leftrightarrow$. Between any two nodes is at most one edge.

A *walk* is a sequence of adjacent nodes. A *path* is a walk in which no node occurs twice. The first node is the *start node*, the last node is the *end node* on a walk, and the remaining nodes are called *internal*. A *cycle* is a walk in which the start and end node are the same node. A walk is *possibly directed*, if every edge is undirected $—$ or directed $\to$ and pointing away from the start node. A walk is *directed* (or causal), if it is possibly directed and contains only directed edges. A walk is *semi-directed*, if it is possibly directed and contains at least one directed edge. If there is a (possibly) directed path from V to W, V is a (*possible*) *ancestor* of W and W is a (*possible*) *descendant* of V. Possible ancestors are also called *anteriors*. $An(\mathbf{W})$ ($pAn(\mathbf{W})$) [$De(\mathbf{W})$ ($pDe(\mathbf{W})$)] denotes all (possible) ancestors [descendants] of a set of nodes $\mathbf{W}$. We write $\mathbf{X} \overset{+}{\sim} \mathbf{Y}$ ($\mathbf{X} \overset{+}{\sim} Z \overset{+}{\sim} \mathbf{Y}$) to mean a walk between two nodes from sets $\mathbf{X}$ and $\mathbf{Y}$ (that also contains node Z). A path is *proper* (relative to a set $\mathbf{X}$), if the start node of the path, but no other node, is in $\mathbf{X}$.

Our results hold for DAGs, AGs, MAGs, RCGs, and CPDAGs. DAGs are the most commonly used causal graphical models. Each DAG encodes a set of probability distributions that are consistent to the DAG [13]. A graph is a *directed acylic graph* (DAG), iff (we use this abbreviation for "if and only if") it contains only $\to$ edges and no directed cycle. Every DAG is an AG and RCG. *Ancestral graphs* (AGs) model ancestral relations, representing all DAGs that have the same ancestral relations, possibly including additional variables that are not included in the AG. A graph is an AG, iff (1) for any edge $A \leftarrow B$ or $A \leftrightarrow B$, A is not an anterior of B, and (2) for any edge $A — B$, there are no edges $A \leftarrow C$, $A \leftrightarrow C$, $B \leftarrow C$ or $B \leftrightarrow C$ [18]. A MAG is a maximal AG. A chain graph (CG) $\mathcal{C}$ is a mixed graph containing no $\leftrightarrow$

edges or semi-directed cycles [10] and represents a set of consistent DAG extensions $\mathcal{D} \in CE(\mathcal{C})$ that are Markov equivalent, i.e., encode exactly the same probability distributions. A graph is a *restricted chain graph* iff (0) it contains no $\leftrightarrow$ edges (1) it contains no possibly directed cycle that contains at least one directed edge (2) for each triple $A \to B - C$, there is an edge between A and C, and (3) every undirected component is chordal. Any CG with $CE(\mathcal{C}) \neq \emptyset$ can be transformed to an equivalent RCG. A completed partially directed acyclic graph (CPDAG) $\mathcal{C}$ represents all Markov equivalent DAGs, i.e., $\mathcal{D}_1, \mathcal{D}_2 \in CE(\mathcal{C})$ if and only if $\mathcal{D}_1$ and $\mathcal{D}_2$ are Markov equivalent. Every CPDAG is an RCG with additional restrictions where directed edges might occur [1, 23].

A node C on a walk π is a *collider*, if both the preceding and succeeding edge on the walk have an arrowhead at C, e.g., $\to C \leftarrow$ or $\leftrightarrow C \leftarrow$. An internal node V on π is called an *almost definite non-collider*, if it occurs as $A \leftarrow V$ or $V \to B$ or $A - V - B$, where A and B are the nodes preceding/succeeding V on π. It is called a *definite non-collider* (or just *non-collider*) in the first two cases or if A and B are not adjacent. An internal vertex on π is said to be of *(almost) definite status*, if it is either a collider or a (almost) definite non-collider on π. A path is said to be of (almost) definite status if all its internal vertices are of (almost) definite status.

A walk π in a mixed graph $\mathcal{G}$ is *active* (d-connected, m-connected, connected) given $\mathbf{Z}$ if every definite non-collider is not in $\mathbf{Z}$ and every collider is in $An(\mathbf{Z})$.

The nodes of a path π in a CG $\mathcal{G}$ also form a path $\pi_{\mathcal{D}}$ in any consistent DAG extension $\mathcal{D} \in CE(\mathcal{G})$, and any definite non-collider on π is a common non-collider on $\pi_{\mathcal{D}}$ in DAG $\mathcal{D}$. So a definite status path π is d-connected in $\mathcal{G}$ if and only if $\pi_{\mathcal{D}}$ is d-connected for all $\mathcal{D} \in CE(\mathcal{G})$ [29]. In an RCG a definite status walk $X \overset{+}{\rightsquigarrow} Y$ exists iff an almost definite status walk $X \overset{+}{\rightsquigarrow} Y$ exists [23]. Every walk in an AG is of almost definite status, since the forbidden configurations are also forbidden in an AG.

A set $\mathbf{Z}$ *separates* (d-separates, m-separates) sets $\mathbf{X}$ and $\mathbf{Y}$ iff there exists no almost definite status walk between any $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ that is active given $\mathbf{Z}$. A set $\mathbf{Z}$ that separates $\mathbf{X}$ and $\mathbf{Y}$ is a *separator* (relative to $\mathbf{X}$ and $\mathbf{Y}$). Traditionally separators in DAGs, RCGs and CPDAGs are called d-separators and separators in AGs and MAGs are called m-separators, but we use a unified definition and thus omit the d- and m- prefixes.

The *augmented graph* $(\mathcal{G})^a$ of a certain AG $\mathcal{G}$ is an undirected graph with the same nodes as $\mathcal{G}$ whose edges are all pairs of nodes that are collider connected in $\mathcal{G}$ (two nodes A, B are called collider connected if there is a path between them on which all nodes except A and B are colliders). Two node sets $\mathbf{X}$ and $\mathbf{Y}$ are m-separated by a node set $\mathbf{Z}$ in $\mathcal{G}$ if and only if $\mathbf{Z}$ is an $\mathbf{X}$-$\mathbf{Y}$ node cut in $(\mathcal{G}_{pAn(\mathbf{X}\cup\mathbf{Y}\cup\mathbf{Z})})^a$ [18]. $(\mathcal{G})^a$ of a DAG $\mathcal{G}$ is called the moral graph $(\mathcal{G})^m$. The construction of the augmented (moral) graph is termed *moralization*.

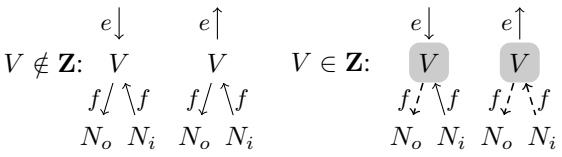

Figure 3: Modified rules for Bayes-Ball in DAGs to test or to find minimal separators. If the nodes V, N_o, N_i are in a given $\mathbf{A} \subseteq \mathbf{V}$ then the Bayes ball goes through the entering top edge e and passes through the node V to nodes N_o (out-node), resp. N_i (in-node). Forbidden passes are marked as dashed arrows. The figure shows all possible combinations of types of entering (e) and leaving edges (f) and considers two cases: $V \notin \mathbf{Z}$ and $V \in \mathbf{Z}$ (gray). The leaving edge f can correspond to the entering edge e in which case the ball might return to the start node of the entering edge, which is called a bouncing ball in the original Bayes-Ball algorithm [19].

3 MINIMAL SEPARATORS IN DAGS WITHOUT MORALIZATION

In this section we show how to test and find minimal d-separators in DAGs directly, i.e., without going through moral graphs, such that their running time is linear in the number of edges in the graph $\mathcal{G}_{An(\mathbf{X}\cup\mathbf{Y})}$. The algorithms are special cases of those presented in Sec. 5, which work for more general classes of causal graphs. The aim to discuss our methods separately for DAGs is twofold: firstly it is interesting to see how they work for this most basic causal structure and secondly, after that it is much easier to understand the main ideas behind the general algorithms.

We start with a set-theoretic characterization of minimal d-separators between sets $\mathbf{X}$ and $\mathbf{Y}$ in a DAG. Let $\mathbf{A}$ be the set $An(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$. Let $\mathbf{X}^\star$ (resp. $\mathbf{Y}^\star$) denote the "closure" of $\mathbf{X}$ ($\mathbf{Y}$), i.e., the set of all nodes V for which there exists a path from $\mathbf{X}$ ($\mathbf{Y}$) to V that only contains nodes of $\mathbf{A}$ and no non-collider in $\mathbf{Z}$. Then a set $\mathbf{Z}$ is a minimal d-separator under the constraint $\mathbf{I} \subseteq \mathbf{Z} \subseteq \mathbf{R}$ if and only if the following *minimality criterion* is satisfied

(a) $\mathbf{X}^\star \cap \mathbf{Y} = \emptyset$, and
(b) $\mathbf{I} \subseteq \mathbf{Z} \subseteq \mathbf{A} \cap \mathbf{R}$, and
(c) $\mathbf{Z} \setminus \mathbf{I}$ is a subset of $\mathbf{X}^\star \cap \mathbf{Y}^\star$.

The first condition says, the set is actually a separator, since it blocks all paths between $\mathbf{X}$ and $\mathbf{Y}$. The second condition is a basic constraint. The third condition verifies that $\mathbf{Z}$ is minimal. For each node $Z \in \mathbf{Z} \setminus \mathbf{I}$ there exists a path from $\mathbf{X}$ to Z and a path from $\mathbf{Y}$ to Z, so Z cannot be removed from $\mathbf{Z}$ without the paths meeting each other and turning $\mathbf{Z}$ into a non-separator. Rather than the first condition the equivalent condition $\mathbf{Y}^\star \cap \mathbf{X} = \emptyset$ could be tested. The sets $\mathbf{X}^\star$ and $\mathbf{Y}^\star$ can be easily found in time $\mathcal{O}(n+m)$ as in the well-known Bayes-Ball algorithm [19], with the difference that (1) when starting in the initial nodes $\mathbf{X}$ (or $\mathbf{Y}$) the ball restricts its tour to nodes from a given set $\mathbf{A}$ and (2) rules for single passing steps are modified. The rules used in our algorithm are shown in Fig. 3: assuming $V \in \mathbf{A}$ a combination (edge type e, V, edge type f, neighbor N) is a "passing" one if and only if $N \in \mathbf{A}$ and if $V \in \mathbf{Z}$ then V is a collider on (e, V, f). Thus, the minimality criterion can be tested in linear time.

For example, the minimality criterion applied to DAG $\mathcal{G}_1$ in Fig. 4, yields that $\mathbf{Z} = \{V_3\}$ is a minimal separator. Set $\mathbf{Z} = \{V_2, V_3\}$ is a separator, but not a minimal one, because it is not a subset of $\mathbf{X}^\star = \{X, V_1, V_2\}$ and $\mathbf{Y}^\star = \{Y, V_3\}$. Similarly, in $\mathcal{G}_2$ (Fig. 4), set $\{V_2, V_3\}$ will be recognized as a minimal separator, while $\mathbf{Z} = \{V_1, V_2, V_3\}$, with $\mathbf{X}^\star = \{X, U, V_1, V_2, V_3\}$ and $\mathbf{Y}^\star = \{Y, V_2, V_3\}$, will be not. For an empty set $\mathbf{Z} = \emptyset$ all nodes are reachable, so $\mathbf{X}^\star = \mathbf{Y}^\star = \mathbf{V}$ in both DAGs and condition (a) shows $\mathbf{Z} = \emptyset$ is not a separator at all.

The main result of this section is that *finding* minimal separators in DAGs can also be done in linear time. The idea is to choose a set like $\mathbf{Z} = (\mathbf{A} \cap \mathbf{R} \cap \mathbf{X}^\star \cap \mathbf{Y}^\star) \cup \mathbf{I}$, which automatically fulfills the minimality criterion.

However, one cannot use this expression directly, since $\mathbf{X}^\star$ and $\mathbf{Y}^\star$ cannot be calculated without knowing $\mathbf{Z}$. But we can perform the intersections one after another and calculate $\mathbf{X}^\star$ with an initial separator $\mathbf{Z}_0$. From this we obtain a new separator, which can be used to calculate $\mathbf{Y}^\star$. This leads to the following algorithm:

function FINDMINSEPINDAG($\mathcal{G}, \mathbf{X}, \mathbf{Y}, \mathbf{I}, \mathbf{R}$)
  Set $\mathbf{A} := An(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$
  Choose a separator $\mathbf{Z}_0 := \mathbf{R} \cap (\mathbf{A} \setminus (\mathbf{X} \cup \mathbf{Y}))$
  Calculate $\mathbf{X}^\star$ using separator $\mathbf{Z}_0$
  Set $\mathbf{Z}_X := \mathbf{Z}_0 \cap \mathbf{X}^\star \cup \mathbf{I}$
  Calculate $\mathbf{Y}^\star$ using separator $\mathbf{Z}_X$
  if $\mathbf{X}^\star \cap \mathbf{Y} \neq \emptyset$ **then return** $\perp$
  return $\mathbf{Z} := \mathbf{Z}_X \cap \mathbf{Y}^\star \cup \mathbf{I}$

For the DAGs in Fig. 4, this approach returns $\mathbf{Z} = \{V_2\}$ in $\mathcal{G}_1$, resp. $\mathbf{Z} = \{V_2, V_3\}$ in $\mathcal{G}_2$.

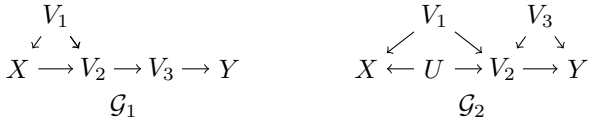

Figure 4: Two DAGs: in $\mathcal{G}_1$ all variables are observed, in $\mathcal{G}_2$ variable U is unobserved. We assume $\mathbf{I} = \emptyset$. For $\mathbf{X} = \{X\}$ and $\mathbf{Y} = \{Y\}$ FINDMINSEPINDAG computes in $\mathcal{G}_1$ separator $\{V_2\}$, resp. in $\mathcal{G}_2$ separator $\{V_2, V_3\}$.

4 GENERALIZED MIN SEPARATORS

Before we present our general algorithms in the next section, below we give some properties of minimal separators, which help to prove correctness of our methods (most proofs are given in the supplementary material).

Let $\mathcal{G}$ be an AG or RCG (for easier understanding, the reader may firstly consider only AGs and then verify that the results are true for RCGs, too). Let $\mathbf{X}, \mathbf{Y}, \mathbf{I}, \mathbf{R}$ be sets of nodes with $\mathbf{I} \subseteq \mathbf{R}$, $\mathbf{R} \cap (\mathbf{X} \cup \mathbf{Y}) = \emptyset$. We only consider minimal separators under the constraint that they contain $\mathbf{I}$, i.e., a separator $\mathbf{Z}$ is minimal if there exists no separator $\mathbf{Z}'$ with $\mathbf{I} \subseteq \mathbf{Z}' \subsetneq \mathbf{Z}$.

First we show that separation rules for walks between nodes $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ become simpler, when the walk only contains nodes of $pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z})$:

Lemma 4.1. *There exists an active almost definite status walk between* $\mathbf{X}$ *and* $\mathbf{Y}$ *given* $\mathbf{Z}$ *iff there exists an almost definite status walk between* $\mathbf{X}$ *and* $\mathbf{Y}$ *s.t. no non-collider is in* $\mathbf{Z}$ *and every collider is in* $pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z})$.

Lemma 4.2. *If* π *is an almost definite status walk* $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}$ *active given* $\mathbf{Z}$*, all nodes of* π *are in* $pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z})$.

Lemma 4.1 and Lemma 4.2 are of course the reason a construction like a moral graph is possible. This implies a corollary, which has also be used in [25] for AGs and [23] for RCGs.

Corollary 4.3 (Ancestry of minimal separators)**.** *Every minimal separator* $\mathbf{Z}$ *is a subset of* $pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$.

This means that $pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}) = pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$ for every minimal separator $\mathbf{Z} \supseteq \mathbf{I}$. So the moral graph for every minimal separator is $(\mathcal{G}_{pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z})})^a = (\mathcal{G}_{pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})})^a$, and all problems involving minimal separators can be solved with corresponding standard algorithms in the undirected, moral graph. However, the moral graph can have $\mathcal{O}(n^2)$ many edges, e.g., if there is one node that is a child of all other nodes. Hence there are no linear time algorithms using the moral graph, and we will use another approach to solve the problems.

Tian, Paz and Pearl show in [21] that a separator $\mathbf{Z}$ in a DAG is minimal, if and only if no node $Z \in \mathbf{Z}$ can be

removed from $\mathbf{Z}$, i.e., iff $\mathbf{Z} \setminus Z$ is not a separator for every $Z \in \mathbf{Z}$. This statement appears trivial, however it is not, since there might be a separator from which removing any single element does not result in a separator while removing two elements together still yields a separator. But due to [21] we know that there is no such case. The result can easily be generalized to AGs or RCGs and then holds for any separator $\mathbf{Z}$, but the proof is lengthy, so we will only state it for separators $\mathbf{Z}$, with $\mathbf{I} \subseteq \mathbf{Z} \subseteq pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$, which only requires a rather short proof.

Lemma 4.4. *A separator $\mathbf{Z}$ between $\mathbf{X}$ and $\mathbf{Y}$, with $\mathbf{I} \subseteq \mathbf{Z} \subseteq pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$, is minimal iff for every $Z \in \mathbf{Z} \setminus \mathbf{I}$ the set $\mathbf{Z} \setminus Z$ is not a separator relative to $(\mathbf{X}, \mathbf{Y})$.*

Now we show that a separator relative to $(\mathbf{X}, \mathbf{Y})$ is minimal, iff all its nodes are reachable from both $\mathbf{X}$ and $\mathbf{Y}$. It is easy to see that this is true in an undirected graph, and was used by [21] and [24] to find minimal separators in the moral graph. But it also holds immediately in an AG or RCG:

Lemma 4.5. *A separator $\mathbf{Z}$ with $\mathbf{I} \subseteq \mathbf{Z}$ is a minimal separator, if and only if for every $Z \in \mathbf{Z} \setminus \mathbf{I}$ there exists an almost definite status walk $\pi : \mathbf{X} \overset{+}{\sim} Z \overset{+}{\sim} \mathbf{Y}$ s.t. every node on π is in $pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$ and every non-collider is not in $\mathbf{Z} \setminus Z$.*

Proof. ($\Rightarrow$): Let $\mathbf{Z}$ be a minimal separator. For every $Z \in \mathbf{Z} \setminus \mathbf{I}$ the set $\mathbf{Z} \setminus Z$ is not a separator due to Lemma 4.4, so there is an almost definite status path (walk) $\pi : \mathbf{X} \overset{+}{\sim} \mathbf{Y}$ open given $\mathbf{Z} \setminus Z$ and blocked by $\mathbf{Z}$, i.e., blocked by Z, so Z is a non-collider on π. Every other non-collider is not in $\mathbf{Z} \setminus Z$ or π was already blocked. From Corollary 4.3 we know $\mathbf{Z} \subseteq pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$, so all nodes of π are in $pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}) = pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$ due to Lemma 4.2.

($\Leftarrow$): Every node in $\mathbf{Z} \setminus \mathbf{I}$ is on an almost definite status walk in $pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$, so $\mathbf{Z} \subseteq pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$. Let $Z \in \mathbf{Z} \setminus \mathbf{I}$ and π be such a walk. Every collider on π is in $pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I}) = pAn(\mathbf{X} \cup \mathbf{Y} \cup (\mathbf{Z} \setminus Z))$, so due to Lemma 4.1 $\mathbf{Z} \setminus Z$ does not separate $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{Z}$ is minimal. □

5 GENERAL ALGORITHMS

In this section we give efficient algorithms to test and find minimal separators in AGs and RCGs and show their correctness based on the results of the previous section. Because a DAG is also an AG and RCG, and the algorithms for DAGs of Sec. 3 are a special case of the algorithms here, this also shows the correctness of the algorithms for DAGs. Due to relations MAG $\subseteq$ AG and CPDAG $\subseteq$ RCG our algorithms also work for MAGs and for CPDAGs.

We start with algorithm REACHABLE that returns all nodes $W \in \mathbf{V}$ that are reachable from a node $X \in \mathbf{X}$ by an almost definite status walk on which all non-colliders are not in $\mathbf{Z}$ and restricted to nodes in $\mathbf{A}$. To traverse graph $\mathcal{G}$ REACHABLE uses the modified rules of the Bayes-Ball algorithm [19] shown in Fig. 5.

function REACHABLE($\mathcal{G}, \mathbf{X}, \mathbf{A}, \mathbf{Z}$)
  Boolean function *pass*(e, V, f, N)
    return $(N \in \mathbf{A}) \wedge$
      (e, V, f) is of almost definite status $\wedge$
      $(V \in \mathbf{Z} \Rightarrow V$ is a collider on $(e, V, f))$
  $\mathbf{Q} := \{(\leftarrow, X) \mid X \in \mathbf{X}\}$
    ▷ queue of pairs (type of edge, node) to visit
  $\mathbf{P} := \mathbf{Q}$ ▷ all processed pairs
  while $\mathbf{Q}$ not empty **do**
    Let (e, V) be the top element in $\mathbf{Q}$
    Remove (e, V) from $\mathbf{Q}$
    for all neighbors N of V **do**
      Let V and N be connected by edge f
      if $(f, N) \notin \mathbf{P} \wedge$ *pass*(e, V, f, N) **then**
        Add (f, N) to $\mathbf{Q}$ and $\mathbf{P}$
  return $\{W \mid (e, W) \in \mathbf{P}$ for any edge type $e\}$

Each node is only visited once from each adjacent edge, so the running time is $\mathcal{O}(n + m)$. Equipped with this method to calculate $\mathbf{X}^\star$ and $\mathbf{Y}^\star$ in mixed graphs, we are ready to give an algorithm which tests the minimality criterion presented in Sec. 3 generalized to AGs and RCGs.

function TESTMINSEP($\mathcal{G}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{I}, \mathbf{R}$)
  if $(\mathbf{I} \not\subseteq \mathbf{Z}) \vee (\mathbf{Z} \not\subseteq \mathbf{R})$ **then return** false
  $\mathbf{A} := pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$
  if $\mathbf{Z} \not\subseteq \mathbf{A}$ **then return** false
  $\mathbf{X}^\star :=$ REACHABLE($\mathcal{G}, \mathbf{X}, \mathbf{A}, \mathbf{Z}$)
  if $\mathbf{X}^\star \cap \mathbf{Y} \neq \emptyset$ **then return** false
  if $\mathbf{Z} \setminus \mathbf{I} \not\subseteq \mathbf{X}^\star$ **then return** false
  $\mathbf{Y}^\star :=$ REACHABLE($\mathcal{G}, \mathbf{Y}, \mathbf{A}, \mathbf{Z}$)
  if $\mathbf{Z} \setminus \mathbf{I} \not\subseteq \mathbf{Y}^\star$ **then return** false
  return true

Proposition 5.1. TESTMINSEP *tests if a set $\mathbf{Z}$ is a minimal separator, with $\mathbf{I} \subseteq \mathbf{Z} \subseteq \mathbf{R}$, in time $\mathcal{O}(n + m)$.*

Proof. TESTMINSEP first checks the basic constraints $\mathbf{I} \subseteq \mathbf{Z} \subseteq \mathbf{R}$ and the constraint $\mathbf{Z} \subseteq pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$ from Corollary 4.3. REACHABLE finds all nodes that are reachable from $\mathbf{X}$ (resp. $\mathbf{Y}$) by almost definite status walks that only contain nodes of $\mathbf{A} = pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I}) = pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z})$ and on which all non-colliders are not in $\mathbf{Z}$. From Lemma 4.2 it follows $\mathbf{Z}$ is a separator if and only if $\mathbf{X}^\star$ contains no node of $\mathbf{Y}$ (alternatively and equivalently $\mathbf{Y}^\star$ contains no node of $\mathbf{X}$).

Then the algorithm returns true iff $\mathbf{Z} \setminus \mathbf{I} \subseteq \mathbf{X}^\star \cap \mathbf{Y}^\star$, which means for each $Z \in \mathbf{Z} \setminus \mathbf{I}$ there exist almost definite status walks $\mathbf{X} \overset{+}{\sim} Z$ and $Z \overset{+}{\sim} \mathbf{Y}$ through $\mathbf{A}$ on which each non-collider is not in $\mathbf{Z}$, i.e., all non-colliders

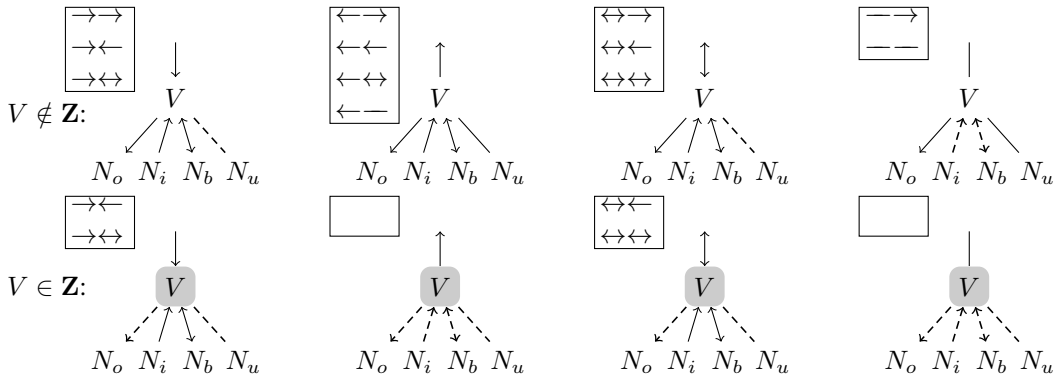

Figure 5: Expanded rules for Bayes-Ball in AGs and RCGs. Assuming all nodes are in $pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$, the rules list (in boxes) all combinations of edge pairs through which the ball is allowed to pass. Similarly as in Fig. 3 the Bayes ball goes through the entering top edge and passes through the node V to the bottom nodes $\{N_o, N_i, N_b, N_u\}$. Forbidden passes are marked as dashed edges. Here, by a pair of edges we mean a top edge and $V \to N_o$ (out-node), resp. $V \leftarrow N_i$ (in-node), $V \leftrightarrow V_b$ (bidirected edge), and $V - N_u$ (undirected edge). We consider two cases: $V \notin \mathbf{Z}$ and $V \in \mathbf{Z}$ (marked gray). The leaving edge can correspond to the entering edge in which case the ball might return to the start node of the entering edge, which is called a bouncing ball in the Bayes-Ball algorithm.

of the combined walk $\mathbf{X} \stackrel{+}{\sim} Z \stackrel{+}{\sim} \mathbf{Y}$ are not in $\mathbf{Z} \setminus Z$ (under the assumption that $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ are disjoint sets), and according to Lemma 4.5 the set $\mathbf{Z}$ is minimal.

The algorithm runs in $\mathcal{O}(n+m)$ as it only uses elementary set operations and calls to REACHABLE. Here we assume the sets are represented as Boolean arrays of size n (or hashsets), so nodes can be added, removed and found in constant time. □

Now, we provide our general method to *find* a minimal separator. We proceed in two steps: First we construct an algorithm, called FINDNEARESTSEP, to compute a separator between $\mathbf{X}$ and $\mathbf{Y}$ that uses nodes closer to $\mathbf{X}$ than to $\mathbf{Y}$ whenever possible. Next, using FINDNEARESTSEP, we give an algorithm to find a minimal separator. An advantage of such a scheme is that already separators generated by FINDNEARESTSEP have some useful properties that we have mentioned in Subsection 1.1 and will discuss in more detail in the next section. Intuitively, both algorithms find separators according to $(\mathbf{X}, \mathbf{Y})$ which are "nearest" to $\mathbf{X}$.

function FINDNEARESTSEP($\mathcal{G}, \mathbf{X}, \mathbf{Y}, \mathbf{I}, \mathbf{R}$)
  $\mathbf{A} := pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$
  $\mathbf{Z}_0 := \mathbf{R} \cap (\mathbf{A} \setminus (\mathbf{X} \cup \mathbf{Y}))$
  $\mathbf{X}^\star :=$ REACHABLE($\mathcal{G}, \mathbf{X}, \mathbf{A}, \mathbf{Z}_0$)
  if $\mathbf{X}^\star \cap \mathbf{Y} \neq \emptyset$ **then return** $\perp$
  return $\mathbf{Z} \cap \mathbf{X}^\star \cup \mathbf{I}$

function FINDMINSEP($\mathcal{G}, \mathbf{X}, \mathbf{Y}, \mathbf{I}, \mathbf{R}$)
  $\mathbf{Z}_X :=$ FINDNEARESTSEP($\mathcal{G}, \mathbf{X}, \mathbf{Y}, \mathbf{I}, \mathbf{R}$)
  $\mathbf{Z}_Y :=$ FINDNEARESTSEP($\mathcal{G}, \mathbf{Y}, \mathbf{X}, \mathbf{I}, \mathbf{Z}_X$)
  if $(\mathbf{Z}_X = \perp) \vee (\mathbf{Z}_Y = \perp)$ **then return** $\perp$
  return $\mathbf{Z}_Y \cap \mathbf{Z}_X \cup \mathbf{I}$

Proposition 5.2. *Algorithm* FINDMINSEP *finds a minimal separator* $\mathbf{Z}$*, with* $\mathbf{I} \subseteq \mathbf{Z} \subseteq \mathbf{R}$*, in time* $\mathcal{O}(n+m)$.

Proof. Since all $\mathbf{Z}_0, \mathbf{Z}_X, \mathbf{Z}_Y$ are enclosed between $\mathbf{I}$ and $\mathbf{A}$, we have $\mathbf{A} = pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}_0) = pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}_X) = pAn(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}_Y)$. Thus we can ignore colliders on the walks between $\mathbf{X}$ and $\mathbf{Y}$ through $\mathbf{A}$ due to Lemma 4.1. FINDMINSEP always returns a set if $\mathbf{X}$, $\mathbf{Y}$ are separable, because $\mathbf{Z}_0$ contains all observable nodes of $\mathbf{A}$ and thus blocks as many paths as possible. When FINDMINSEP returns a set, it is a separator with $\mathbf{I} \subseteq \mathbf{Z} \subseteq \mathbf{R}$. The test $\mathbf{X}^\star \cap \mathbf{Y} \neq \emptyset$ ensures that $\mathbf{Z}_0$ is a separator, and all nodes removed for $\mathbf{Z}_X$ ($\mathbf{Z}_Y$) are not reachable from $\mathbf{Y}$ ($\mathbf{X}$), so they cannot block a walk and the set remains a separator after their removal. All remaining nodes are reachable from both $\mathbf{X}$ and $\mathbf{Y}$, so $\mathbf{Z}$ is minimal according to Lemma 4.5.

The algorithm runs in $\mathcal{O}(n+m)$ as it only uses elementary set operations and calls to REACHABLE. □

6 APPLICATIONS

Here we apply our algorithms to the problems introduced in Subsection 1.1.

6.1 DAG consistency testing

The Causal Markov Condition says a variable V is conditionally independent of $\mathbf{V} \setminus (Pa(V) \cup De(V))$ given its parents $Pa(V)$. These n independence statements can be tested on the joint probability distribution of the data,

but they involve large sets of variables, which are statistically hard to verify. It is thus desirable to reduce the number of variables in each independence test.

This can be accomplished by performing pairwise conditional tests on a topological ordering $V_1, \ldots, V_n$ of the variables $\mathbf{V}$, with $V_j \notin An(V_i)$ for $i < j$. If there is no edge $V_i \to V_j$, then V_i should be conditionally independent of V_j given $Pa(V_j)$. This yields $n^2/2$ tests between single variables.

In [14] it has been shown that one can replace the parents $Pa(V_j)$ in the test by any set $\mathbf{Z}_{ij}$ such that all nodes of $\mathbf{Z}_{ij}$ are closer to X_j than to X_i. So minimal separators $\mathbf{Z}_{ij} \subseteq \mathbf{R}_{ij} = \{X_k \in \mathbf{V} \mid d(X_k, X_j) < d(X_i, X_j)\}$ can be used in the test (where the function d measures the length of the shortest path between two nodes).

An empirical analysis on random DAGs has found that depending on the number of edges using such minimal separators requires up to 90% fewer conditioning variables than using the parents in the graph [25].

However, finding a minimal separator for all pairs of nodes using moral graphs requires time $\mathcal{O}(n^4)$. Using our linear time algorithms this time reduces to $\mathcal{O}(n^2 m)$, which is more feasible.

6.2 Adjustment sets

For a DAG [24, 25], adjustment amenable MAG [24, 25], CPDAG [17, 23] or RCG [23] $\mathcal{G}$ the following criterion is satisfied by a set $\mathbf{Z}$ iff $\mathbf{Z}$ is an adjustment set. Let $PCP(\mathbf{X}, \mathbf{Y}) = \{W \in \mathbf{V} \setminus \mathbf{X} \mid W$ lies on a proper possible causal path from $\mathbf{X}$ to $\mathbf{Y}\}$.

Definition 6.1 (Proper back-door graph [25])**.** *Let $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$ be pairwise disjoint subsets of variables. The* proper back-door graph*, denoted as $\mathcal{G}^{pbd}_{\mathbf{XY}}$, is obtained from $\mathcal{G}$ by removing every edge from a node in $\mathbf{X}$ to a node in $PCP(\mathbf{X}, \mathbf{Y})$.*

For an example of the proper back-door graph see Fig. 1.

Definition 6.2 (Constructive back-door criterion (CBC) [25])**.** *Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$ be pairwise disjoint subsets of variables. The set $\mathbf{Z}$ satisfies the CBC relative to $(\mathbf{X}, \mathbf{Y})$ if (a) $\mathbf{Z} \subseteq \mathbf{V} \setminus pDe(PCP(\mathbf{X}, \mathbf{Y}))$ and (b) $\mathbf{Z}$ separates $\mathbf{X}$ and $\mathbf{Y}$ in the proper back-door graph $\mathcal{G}^{pbd}_{\mathbf{XY}}$.*

The CBC makes it easy to test or find a minimal adjustment set. Every minimal adjustment set is a minimal separator in the proper back-door graph $\mathcal{G}^{pbd}_{\mathbf{XY}}$ under the constraint $\mathbf{R}' = \mathbf{R} \setminus pDe(PCP(\mathbf{X}, \mathbf{Y}))$. The necessary set operations and construction of the proper backdoor graph can be done in linear time, but finding a minimal separator using existing algorithms based on the moral graph [21, 24, 25] requires quadratic time. Thus, the best known algorithms for minimal adjustment sets so far take quadratic time. Using our algorithms we conclude:

Corollary 6.3. *Testing and finding a minimal adjustment set in a DAG, CPDAG, RCG or in an adjustment amenable MAG can be done in linear time.*

Separators constructed by our algorithms presented in Sec. 5 have certain properties useful to obtain valid adjustment sets which, besides minimality, satisfy some further conditions. Below we show that our methods allow to tackle some issues of statistical significance. Specifically, we discuss selections of covariates for estimating total effect that minimize the asymptotic variance of the estimators.

In [27] VanderWeele and Shpitser recommended the following strategy for determining which covariates to control for as confounders if a causal structure $\mathcal{G}$ is known: Suppose that for some set $\mathbf{Z}_1$ which d-separates $\mathbf{X}$ and $\mathbf{Y}$, and that under an ordering of the elements of $\mathbf{Z}_1$, $V_1, \ldots, V_t$, for some s and $\mathbf{Z}_2 = \{Z_1, \ldots, Z_s\}$, set $\mathbf{X} \cup \mathbf{Z}_2$ d-separates $\mathbf{Y}$ and Z_i for $i = s+1, \ldots, t$, then $\mathbf{Z}_2$ d-separates $\mathbf{Y}$ and $\mathbf{X}$. This allows to discard covariates that may be a cause of treatment $\mathbf{X}$ but unrelated to the outcome $\mathbf{Y}$ and thus $\mathbf{Z}_2$ should be preferred to $\mathbf{Z}_1$. In [8, 15] Henckel et al. provide a more general criterion to compare two adjustment sets:

$$\begin{aligned} &\mathbf{Z}_1 \setminus \mathbf{Z}_2 \text{ is } d\text{-separated from } \mathbf{Y} \text{ given } \mathbf{Z}_2 \cup \mathbf{X} \text{ and} \\ &\mathbf{Z}_2 \setminus \mathbf{Z}_1 \text{ is } d\text{-separated from } \mathbf{X} \text{ given } \mathbf{Z}_1, \end{aligned} \tag{1}$$

and prove that selecting covariates according to this criterion decreases the asymptotic variance of the total effect estimate in the multivariate linear Gaussian models.

Recall, that the total effect of $\mathbf{X} = \{X_1, \ldots, X_k\}$ on $\mathbf{Y} = \{Y_1, \ldots, Y_\ell\}$ in a linear structural causal model is an $\ell \times k$ matrix such that the (j, i)-th element $(\tau_{\mathbf{yx}})_{ji} = \frac{\partial}{\partial x_i} E[Y_j | do(x_1, \ldots, x_k)]$. If $\mathbf{Z}$ is a valid adjustment set relative to $(\mathbf{X}, \mathbf{Y})$ in $\mathcal{G}$ then $\tau_{\mathbf{yx}} = \tau^{\mathbf{z}}_{\mathbf{yx}}$, where $(\tau^{\mathbf{z}}_{\mathbf{yx}})_{ji}$ is the coefficient of X_i in the linear regression of Y_j on $\mathbf{X}$ and $\mathbf{Z}$. Let $\hat{\tau}^{\mathbf{z}}_{\mathbf{yx}}$ denote the ordinary least squares estimate of $\tau^{\mathbf{z}}_{\mathbf{yx}}$. The goal is to find $\mathbf{Z}$ that for every j, i minimizes the asymptotic variance of the estimator $(\hat{\tau}^{\mathbf{z}}_{\mathbf{yx}})_{ji}$. In general, if $\hat{\theta}_N$, with $N = 1, 2, 3, \ldots$ is a sequence of estimators, then the *asymptotic variance* of $\hat{\theta}$ is defined as $aVar[\hat{\theta}] = \lim_{N\to\infty}(N \cdot \hat{\theta}_N)$, whenever this limit exists. Below we write, for short, $aVar[\hat{\tau}^{\mathbf{z}_2}_{\mathbf{yx}}] \leq aVar[\hat{\tau}^{\mathbf{z}_1}_{\mathbf{yx}}]$ if $aVar[(\hat{\tau}^{\mathbf{z}_2}_{\mathbf{yx}})_{ji}] \leq aVar[(\hat{\tau}^{\mathbf{z}_1}_{\mathbf{yx}})_{ji}]$ for all j and i.

Lemma 6.4 ([8, 15])**.** *Let for given $\mathbf{X}, \mathbf{Y}$ in a DAG $\mathcal{G}$ two valid adjustment sets $\mathbf{Z}_1, \mathbf{Z}_2$ satisfy condition* (1)*. Then $aVar[\hat{\tau}^{\mathbf{z}_2}_{\mathbf{yx}}] \leq aVar[\hat{\tau}^{\mathbf{z}_1}_{\mathbf{yx}}]$.*

If the MAG is not adjustment amenable for $\mathbf{X}$ and $\mathbf{Y}$, no adjustment set exists. Unfortunately the test for adjustment amenability requires $\mathcal{O}(k(n+m))$ time for k children of $\mathbf{X}$.

Basically the asymptotic variance of the estimator via adjustment decreases when nodes close to $\mathbf{Y}$ are included in the set and worsens with nodes close to $\mathbf{X}$. As algorithm FINDNEARESTSEP includes all nodes reachable from $\mathbf{Y}$, it is well-suited to find optimal adjustment sets.

Proposition 6.5. *For $\mathbf{X}, \mathbf{Y}$ in a DAG $\mathcal{G}$ and a valid adjustment set $\mathbf{Z}$, with $\mathbf{I} \subseteq \mathbf{Z} \subseteq An(\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I})$, $\mathbf{Z}_2 :=$ FINDNEARESTSEP$(\mathcal{G}^{pbd}_{\mathbf{XY}}, \mathbf{Y}, \mathbf{X}, \mathbf{I}, \mathbf{Z})$ computes in time $\mathcal{O}(n+m)$ a valid adjustment set $\mathbf{Z}_2$, such that for every valid adjustment $\mathbf{Z}_1 \subseteq \mathbf{Z}$, with $\mathbf{I} \subseteq \mathbf{Z}_1$, $aVar[\hat{\tau}^{\mathbf{z}_2}_{\mathbf{yx}}] \leq aVar[\hat{\tau}^{\mathbf{z}_1}_{\mathbf{yx}}]$.*

When all nodes in the graph are observed, the algorithm returns an optimal adjustment set. The asymptotic runtime is the same as the closed form expression $Pa(PCP(\mathbf{X}, \mathbf{Y})) \setminus De(PCP(\mathbf{X}, \mathbf{Y}))$ given by [8, 15] for an optimal adjustment set.

In the presence of unobserved nodes our algorithms can still find an optimal subset of another adjustment set. This improves upon the running time of the pruning algorithm of [8, 15] in the case of adjustment sets consisting of ancestors of $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{I}$. However, further research is needed to develop a linear time algorithm to handle the case of unobserved ancestors. For example in the DAG $X \to Y \leftarrow U \to Z$ with a latent node U the optimal adjustment set is $\{Z\}$ and not the minimal empty set found by our algorithms. In many graphs with unobserved nodes no optimal adjustment set exists [8, 15].

6.3 Instrumental variables

In linear causal models the causal effect can be identified using (conditional) instrumental variables.

Definition 6.6 (Conditional Instrument [13])**.** *Z is said to be a* conditional instrument *relative to $X \to Y$, if there exists a set $\mathbf{W} \subseteq \mathbf{R}$ such that*

(a) $\mathbf{W}$ *does not d-separate Z and X,*
(b) $\mathbf{W}$ *d-separates Z and Y in $\mathcal{G}_c = \mathcal{G} \setminus (X \to Y)$, and*
(c) $\mathbf{W}$ *contains no descendant of Y.*

In [26] we have shown that given variables X, Y and Z it is NP-complete to decide if Z is a conditional instrument, because finding the set $\mathbf{W}$ is NP-complete. Nevertheless, if there exists any conditional instrument for given variables X, Y, there also exists a (so called ancestral) conditional instrument Z' with its $\mathbf{W}' \subseteq An(Y, Z')$, and this $\mathbf{W}'$ can be found in $\mathcal{O}(nm)$ [26]. Our new algorithm can find such a set $\mathbf{W}'$ in linear time $\mathcal{O}(n+m)$.

A set $\mathbf{W} \subseteq \mathbf{R} \cap An(Y, Z)$ is a *nearest separator* relative to (Y, Z) if (i) $\mathbf{W}$ d-separates Y and Z (ii) for all $X \in An(Y \cup Z) \setminus \{Y, Z\}$ and any path π connecting X and Z in the moral graph $(\mathcal{G}_{An(Y \cup Z)})^m$ if there exists $\mathbf{W}' \subseteq \mathbf{R} \cap An(Y, Z)$ that separates Y and Z given $\mathbf{W}'$ and that does not contain a node of π then $\mathbf{W}$ does not contain a node of π either [26, 22]. A nearest separator relative to (Y, Z) can be used as $\mathbf{W}$ in Definition 6.6 to test if Z is an ancestral conditional instrument [26] relative to $X \to Y$ (note that the nearest separator does not depend on X). It can also be used for a set of instrumental variables [22].

Nearest separators should not contain unnecessary nodes similar to minimal separators. However, the rules deciding which nodes are unnecessary are different, so the concepts of nearest and minimal separators are orthogonal. There exist minimal separators that are not nearest separators like a set $\{W\}$ in the DAG $Y \to X \to W \to Z$, and there exist nearest separators that are not minimal, like $\{W_1, W_2\}$ in the DAG $W_1 \to Y \to W_2 \to Z$.

Using algorithm FINDNEARESTSEP we get the following

Proposition 6.7. *Given a DAG $\mathcal{G}$, nodes Y and Z,* FINDNEARESTSEP$(\mathcal{G}, Y, Z, \emptyset, \mathbf{R})$ *finds in time $\mathcal{O}(n+m)$ a nearest separator $\mathbf{W} \subseteq An(Y, Z)$ according to (Y, Z), if Y and Z are d-separable, or it returns $\perp$ otherwise.*

7 CONCLUSIONS AND DISCUSSION

In this paper we show that testing and finding a minimal separator in DAGs, resp. in MAGs, AGs, CPDAGs and RCGs can be done in linear time. Our algorithms are implemented in the open-source software `dagity` [20] and we hope they allow to improve the efficiency of more complex graphical causal methods, that use d- or m-separations as graphical primitives.

It is an open problem if such an algorithm exists for common chain graphs. Every CG $\mathcal{C}$, with $CE(\mathcal{C}) \neq \emptyset$ can be transformed to an equivalent RCG for which such a separator can be computed. However, such a transformation needs time $\mathcal{O}(k^2 m) \leq \mathcal{O}(n^4)$, where k describes the maximum degree of nodes in a given CG [23].

We have considered minimal separators under the constraint $\mathbf{I} \subseteq \mathbf{Z}$, such that there should not exist any separator $\mathbf{Z}' \subsetneq \mathbf{Z}$ in the class of all separators containing $\mathbf{I}$. One could also investigate *strongly-minimal* separators, i.e., such that $\mathbf{I} \subseteq \mathbf{Z}$ and no separator $\mathbf{Z}' \subsetneq \mathbf{Z}$ exists. If $\mathbf{X}$ and $\mathbf{Y}$ are separable, then a minimal $\mathbf{Z}$ always exists and can be found in linear time. But a strongly-minimal $\mathbf{Z}$ might not exist at all, e.g., in the DAG $X \to I \leftarrow V \to Y$ with $\mathbf{I} = \{I\}$, set $\mathbf{Z} = \{I, V\}$ is the only separator satisfying $\mathbf{I} \subseteq \mathbf{Z}$, but the empty set is a separator as well. If a strongly-minimal separator exists, it is also a minimal one. However, in [25] it is shown that it is NP-hard to find a strongly-minimal separating set. Hence attempting to design efficient algorithms for strongly-minimal separators would be futile.

References

[1] ANDERSSON, S. A., MADIGAN, D., PERLMAN, M. D., ET AL. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics 25*, 2 (1997), 505–541.

[2] ANGRIST, J. D., IMBENS, G. W., AND RUBIN, D. B. Identification of causal effects using instrumental variables. *Journal of the American Statistical Association 91*, 434 (1996), 444–455.

[3] BOWDEN, R., AND TURKINGTON, D. *Instrumental variables*. Cambridge University Press, 1984.

[4] BRITO, C. *Graphical Methods for Identification in Structural Equation Models*. PhD Thesis, Dept. of Comp. Sc., Univ. of California, Los Angeles, 2004.

[5] BRITO, C., AND PEARL, J. Generalized instrumental variables. In *Proc. 18th Conf. on Uncertainty in Artificial Intelligence, UAI* (2002), pp. 85–93.

[6] DAWID, P. Conditional independence in statistical theory. *Journal of the Royal Statistical Society 41*, 1 (1979), 1–31.

[7] HAUSER, A., AND BÜHLMANN, P. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research 13*, Aug (2012), 2409–2464.

[8] HENCKEL, L., PERKOVIĆ, E., AND MAATHUIS, M. H. Graphical criteria for efficient total effect estimation. In *European Causal Inference Meeting, EuroCIM* (2019).

[9] IMBENS, G. Instrumental variables: An econometrician's perspective. *Statistical Science 29*, 3 (2014), 323–358.

[10] LAURITZEN, S., AND WERMUTH, N. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics 17* (1989), 31–57.

[11] MEEK, C. Causal inference and causal explanation with background knowledge. In *Proc. 11th Conf. on Uncertainty in Artificial Intelligence, UAI* (1995), Morgan Kaufmann, pp. 403–410.

[12] PEARL, J. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, San Mateo, 1988.

[13] PEARL, J. *Causality*. Cambridge University Press, 2009.

[14] PEARL, J., AND MESHKAT, P. Testing regression models with fewer regressors. In *Proc. of the 7th Int. Workshop on Artificial Intelligence and Statistics, AISTATS* (1999), Morgan Kaufmann, pp. 255–259.

[15] PERKOVIĆ, E. *Graphical characterizations of adjustment sets*. PhD Thesis, ETH Zurich, 2018.

[16] PERKOVIĆ, E., KALISCH, M., AND MAATHUIS, M. Interpreting and using CPDAGs with background knowledge. In *Proc. 33rd Conf. on Uncertainty in Artificial Intelligence* (2017), p. .

[17] PERKOVIĆ, E., TEXTOR, J., KALISCH, M., AND MAATHUIS, M. Complete graphical characterization and construction of adjustment sets in Markov equivalence classes of ancestral graphs. *Journal of Machine Learning Research 18*, 220 (2018), 1–62.

[18] RICHARDSON, T., AND SPIRTES, P. Ancestral graph Markov models. *Annals of Statistics 30* (2002), 927–1223.

[19] SHACHTER, R. D. Bayes-ball: The rational pastime. In *Proc. 14th Conf. on Uncertainty in Artificial Intelligence, UAI* (1998), Morgan Kaufmann, pp. 480–487.

[20] TEXTOR, J., VAN DER ZANDER, B., GILTHORPE, M. S., LIŚKIEWICZ, M., AND ELLISON, G. T. Robust causal inference using directed acyclic graphs: the R package 'dagitty'. *International Journal of Epidemiology 45*, 6 (2016), 1887–1894.

[21] TIAN, J., PAZ, A., AND PEARL, J. Finding minimal d-separators. Tech. Rep. R-254, Univ. of California, Los Angeles, 1998.

[22] VAN DER ZANDER, B., AND LIŚKIEWICZ, M. On searching for generalized instrumental variables. In *Proc. 19th Int. Conf. on Artificial Intelligence and Statistics, AISTATS* (2016), pp. 1214–1222.

[23] VAN DER ZANDER, B., AND LIŚKIEWICZ, M. Separators and Adjustment Sets in Markov Equivalent DAGs. In *Proc. 30th AAAI Conf. on Artificial Intelligence* (2016), pp. 3315–3321.

[24] VAN DER ZANDER, B., LIŚKIEWICZ, M., AND TEXTOR, J. Constructing separators and adjustment sets in ancestral graphs. In *Proc. 30th Conf. on Uncertainty in Artificial Intelligence, UAI* (2014), AUAI Press, pp. 907–916.

[25] VAN DER ZANDER, B., LIŚKIEWICZ, M., AND TEXTOR, J. Separators and adjustment sets in

causal graphs: Complete criteria and an algorithmic framework. *Artificial Intelligence 270* (2019), 1–40.

[26] VAN DER ZANDER, B., TEXTOR, J., AND LIŚKIEWICZ, M. Efficiently finding conditional instruments for causal inference. In *Proc. 24th Inter. Joint Conf. on Artificial Intelligence (IJCAI)* (2015), pp. 3243–3249.

[27] VANDERWEELE, T. J., AND SHPITSER, I. A new criterion for confounder selection. *Biometrics 67*, 4 (2011), 1406–1413.

[28] VERMA, T., AND PEARL, J. Causal networks: semantics and expressiveness. In *Proc. 4th Conf. on Uncertainty in Artificial Intelligence, UAI* (1988), pp. 69–78.

[29] ZHANG, J. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research 9* (2008), 1437–1474.