On Searching for Generalized Instrumental Variables

Benito van der Zander **Maciej Liśkiewicz**
Institute of Theoretical Computer Science, University of Lübeck, Germany
{benito,liskiewi}@tcs.uni-luebeck.de

Abstract

Instrumental Variables are a popular way to identify the direct causal effect of a random variable X on a variable Y. Often no single instrumental variable exists, although it is still possible to find a set of generalized instrumental variables (GIVs) and identify the causal effect of all these variables at once. Till now it was not known how to find GIVs systematically or even test efficiently, if given variables satisfy GIV conditions. We provide fast algorithms for searching and testing restricted cases of GIVs. However, we prove that in the most general case it is NP-hard to verify if given variables fulfill the conditions of a general instrumental sets

1 Introduction

Structural Equation Models (SEMs) are a widely applied tool in the social sciences and economics. They are used to encode and analyze the causal and statistical relationships between the random variables of interest whose interaction is assumed to be linear (Bollen, 1989; Duncan, 1975). In this paper we study the problem of estimating the strength of cause-effects relationships in linear models from observational data and the structure of the model. This problem, known as *the identification problem* (Fisher, 1966), plays a fundamental role in theory and practice of SEMs. Though some partial solutions are given, in general, the problem remains still open.

We investigate graphical methods to this problem. A primary benefit of such approach is that it provides an elegant framework for analyzing linear models by encoding the structure of the model as a directed acyclic graph (DAG). This allows one to attack the identification problem using techniques developed in computer science.

One of the most popular methods to identify single parameters in linear models is based on the concept of instrumental variables (IV) (Bowden and Turkington, 1984). Since the methods provide sufficient but not necessary criteria, they are often not applicable, even if the parameters are uniquely identified. Brito (2004, 2010) and Brito and Pearl (2002a) have generalized this method to allow the identification of multiple parameters simultaneously, introducing *instrumental sets*. However, an important barrier to the application of this method is of algorithmic nature: So far, it was not clear whether such instrumental sets can be found efficiently. Moreover, until now no results have been known demonstrating that searching for instrumental sets is hard either.

Recently, other methods providing sufficient graphical criteria for the parameter identification have been proposed (Tian, 2007; Brito and Pearl, 2006, 2002b; Chen et al., 2014). Though it is not clear whether the methods have more identification power than the instrumental set based ones, their great advantage is that they lend themselves well to algorithmic implementations. In our paper we show that many variants of instrumental sets can be constructed efficiently as well.

We analyze three layers of instrumental sets, from very simple ones introduced by Pearl (2009) and extended by Brito (2010) to the most general ones defined by Brito and Pearl (2002a) and Brito (2004). We provide efficient algorithms to find instrumental sets and to test given sets for being instrumental sets on the simplest level, as well as to test them on the middle level. We show that testing on the most general layer is NP-complete, however, we describe an algorithm that runs in polynomial time under the assumption that the size of the set is bounded by a constant.

In the next section we provide graph preliminaries and define the identification problem in linear models formally. Section 3 discusses IV methods for identifica-

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

tion. Section 4 presents our constructive results, while Section 5 lists the algorithms themselves. Finally in Section 6 we prove the NP-hardness of the general case.

2 Preliminaries

Graphs, d-Separation, Paths. We denote sets by bold upper case letters ($\mathbf{S}$), and sometimes abbreviate singleton sets as $S = \{S\}$. Graphs are written calligraphically ($\mathcal{G}$), and variables in upper-case (X).

We consider graphs $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ with nodes (vertices, variables) $\mathbf{V}$ and directed ($A \to B$) and bidirected ($A \leftrightarrow B$) edges $\mathbf{E}$. Nodes linked by an edge are *adjacent*. A *path* of length ℓ is a node sequence $A_1, \ldots, A_{\ell+1}$, in which no A_i occurs more than once, such that there exists an edge sequence $E_1, E_2, \ldots, E_\ell$ for which every edge E_i connects A_i, A_{i+1}. Then A_1 is called the *start node* and $A_{\ell+1}$ the *end node* of the path. We use the terms *child*, *parent*, *ancestor* and *descendant* to describe node relationships in graphs in the same way as in Pearl (2009); in this convention, every node is an ancestor (but not a parent) and a descendant (but not a child) of itself. For a node set $\mathbf{Y}$ we denote by $An(\mathbf{Y})$ the set of all ancestors of nodes in $\mathbf{Y}$. For a path π we denote by $\pi[A_i \sim A_j]$ the subpath of π consisting of the nodes $A_i, A_{i+1}, \ldots, A_j$.

A node V on a path π is called a *collider* if two arrowheads of π meet at V, e.g. if π contains $U \to V \leftarrow Q$. There can be no collider if π is shorter than 2. Two nodes U, V are called *d-connected* by a set $\mathbf{W}$ if there is a path π between them on which every node that is a collider is in $An(\mathbf{W})$ and every node that is not a collider is not in $\mathbf{W}$. Then π is called a d-connecting path. If U, V are d-connected by the empty set, we simply say they are d-connected. If U, V are not d-connected by $\mathbf{W}$, we say that $\mathbf{W}$ *d-separates* them or *blocks* all paths between them. Two node sets $\mathbf{X}, \mathbf{Y}$ are d-separated by $\mathbf{W}$ if all their nodes are pairwise d-separated by $\mathbf{W}$, which we denote as $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{W})_{\mathcal{G}}$. Otherwise, if they are not d-separated by $\mathbf{W}$, we write $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} \mid \mathbf{W})_{\mathcal{G}}$.

Let $\pi_1, \ldots, \pi_k$ be unblocked paths connecting the variables $Z_1, \ldots, Z_k$ to the variables $X_1, \ldots, X_k$, respectively. We say that the paths $\pi_1, \ldots, \pi_k$ are *incompatible* if for all $1 \leq i < j \leq k$, variable Z_j does not appear in path π_i; and, if paths π_i and π_j have a common variable V, then both $\pi_i[V \sim X_i]$ and $\pi_j[Z_j \sim V]$ point to V. This definition implies that it is not possible to rearrange the edges of incompatible paths to create new paths between the same nodes.

Parameter Identification in Linear Models. A linear model over random variables $V_1 \ldots, V_n$ is defined by a set of equations of the form

$$V_j = \sum_i c_{ji} V_i + \varepsilon_j, \quad j = 1, \ldots, n. \tag{1}$$

$$\begin{aligned} Z_1 &= \varepsilon_1 \\ Z_2 &= \varepsilon_2 \\ X_1 &= a_1 Z_1 + a_2 Z_2 + \varepsilon_3 \\ X_2 &= b_1 Z_1 + b_2 Z_2 + \varepsilon_4 \\ Y &= c_1 X_1 + c_2 X_2 + \varepsilon_5 \end{aligned}$$

$$\begin{aligned} Cov(\varepsilon_3, \varepsilon_5) &= \alpha_1 \neq 0 \\ Cov(\varepsilon_4, \varepsilon_5) &= \alpha_2 \neq 0 \end{aligned}$$

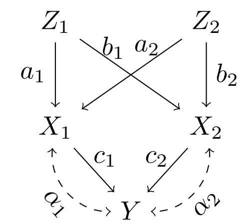

Figure 1: The linear model and its causal graph.

Parameters c_{ji} are called *path coefficients* and they describe direct causal effects of V_i on V_j. In this paper we consider only recursive models, i.e. we assume that for all $i \geq j$ we have $c_{ji} = 0$. Thus, in particular, in Eq. (1) we sum over all $i < j$. Values ε_j represent error terms and are assumed to be normally distributed. We denote the matrix of coefficients c as $C = [c_{ji}]$, the error covariance matrix as $\Psi = [Cov(\varepsilon_i, \varepsilon_j)]$ and the covariance matrix over the observed variables as $\Sigma = [Cov(V_i, V_j)]$. The parameters of the linear system are the non-zero entries in C and Ψ.

The structure of a linear model over $V_1 \ldots, V_n$ can be represented by a DAG $\mathcal{G}$, called a *causal graph*, whose nodes $\mathbf{V}$ correspond to the model's variables and edges indicate the non-zero parameters of the model; $\mathcal{G}$ contains a directed edge $V_i \to V_j$ if V_i appears in Eq. (1) on the right hand side of the equation for V_j with $c_{ji} \neq 0$ and $\mathcal{G}$ contains a bidirected edge $V_i \leftrightarrow V_j$ (displayed in dashed style) if $Cov(\varepsilon_i, \varepsilon_j) \neq 0$. The causal graph can be completed with edge labeling representing the parameters. For an exemplary linear model and its causal graph, see Fig. 1.

Given a structure of a linear model and its parameters represented as C and Ψ, the covariance matrix Σ of the model is given by the formula (Bollen, 1989):

$$\Sigma = (I - C)^{-1} \Psi ((I - C)^{-1})^T. \tag{2}$$

The identification problem consists of recovering the parameters C given the observed covariance matrix Σ and the structure of the model, given e.g. as a causal graph. To solve the identification problem one can attempt to search for a solution of Eq. (2) for given Σ with unknowns C which is independent of the unobserved error correlation Ψ.

If, given Σ and a causal graph, there exists a unique solution c_{ji} satisfying Eq. (2), independent of Ψ, then the path coefficient c_{ji} is said to be *identified*; otherwise it is said to be *nonidentifiable*. If every parameter of the model is identified then we say that *the model is identified*.

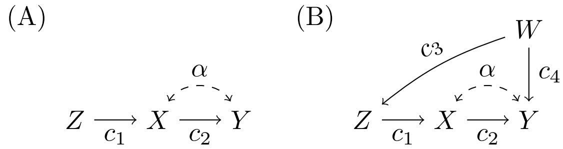

Figure 2: (A) The classical IV and (B) an example of a conditional IV Z given W.

3 Identification with IV Methods

The instrumental variable (IV) approach is one of the most popular methods to identify a single parameter $X \overset{c}{\to} Y$ in linear models. Expressed in graphical language, Z is an IV relative to $X \to Y$ in a graph $\mathcal{G}$ if Z is not d-separated from X and Z is d-separated from Y in the graph obtained from $\mathcal{G}$ by deleting the edge $X \to Y$ (for an example see Fig. 2(A)). If such an instrument exists, then the parameter c can be estimated as $c = Cov(Z, Y)/Cov(Z, X)$. The conditions of IV are sufficient but not necessary to identify the parameter $X \overset{c}{\to} Y$. Pearl (2009) gave a generalization of the method through the use of conditioning. Variable Z is said to be a *conditional instrument* relative to $X \to Y$, if there exists a set $\mathbf{W}$ of non-descendants of Y such that $\mathbf{W}$ does not d-separate Z and X, and $\mathbf{W}$ d-separates Z and Y in the graph obtained from $\mathcal{G}$ by deleting the edge $X \to Y$ (see Fig. 2(B)). When a conditional variable Z given $\mathbf{W}$ is found, the causal effect of X on Y is identified and given by $c = Cov(Z, Y \mid \mathbf{W})/Cov(Z, X \mid \mathbf{W})$.

However, the graphical characterization of conditional instruments does not indicate how to find Z and $\mathbf{W}$. A direct implementation of the conditions requires an exponential running time. This has been regarded as one of the major drawbacks of this approach. Recently van der Zander et al. (2015) have shown that this barrier can be overcome: they give efficient and simple algorithms to find conditional instruments in causal graphs.

The IV method was further generalized to allow identification of multiple parameters simultaneously (Brito and Pearl, 2002a; Brito, 2010). Such an approach can be applied in cases when the linear model includes the equation $Y = c_1X_1 + \ldots + c_kX_k + \varepsilon$, but repeated application of a method for single parameter identification is not possible, like e.g. in the model in Fig. 1.

Let Y be a fixed variable and let $X_1 \overset{c_1}{\to} Y, \ldots, X_k \overset{c_k}{\to} Y$ be edges representing directed causes of Y in the causal diagram $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ of a linear model. Let $\overline{\mathcal{G}}$ be the graph obtained from $\mathcal{G}$ by deleting edges $X_1 \to Y, \ldots, X_k \to Y$ from $\mathcal{G}$. Brito (2010) proposed the following simple generalization of the IV which allows identification of parameters $c_1, \ldots, c_k$ simultaneously.

Definition 3.1 (Brito (2010))**.** *The set* $\mathbf{Z}$ *is said to be a* simple instrumental set *relative to* $\mathbf{X}$ *and* Y *in* $\mathcal{G}$ *if for a permutation* $Z_1, \ldots, Z_k$ *of* $\mathbf{Z}$ *and a permutation* $X_1, \ldots, X_k$ *of* $\mathbf{X}$ *it is true:*

(a) There exist unblocked paths $\pi_1, \ldots, \pi_k$ *connecting* $Z_1, \ldots, Z_k$ *to* $X_1, \ldots, X_k$*, resp., s.t. the paths are incompatible.*
(b) The variables Z_i *are d-separated from* Y *in* $\overline{\mathcal{G}}$*.*

Using Wright's method of path coefficients (Wright, 1934) Brito proves that if we can find variables $\{Z_1, \ldots, Z_k\}$ satisfying the conditions above, then the parameters $c_1, \ldots, c_k$ are identified[1], and can be computed by solving the following system of linear equations:

$$\begin{aligned} &\rho_{Z_1,Y} = a_{11}c_1 + \ldots + a_{1k}c_k \\ &\ldots \\ &\rho_{Z_k,Y} = a_{k1}c_1 + \ldots + a_{kk}c_k, \end{aligned}$$

where $a_{ij} = \rho_{Z_i,X_j}$ and $\rho_{Z,Y}$ denotes the correlation coefficient of Z of Y. Thus, each coefficient of the system of equations above can be estimated from data and the solution of the equations provides the parameter values for $c_1, \ldots, c_k$.

It is easy to see that variables Z_1 and Z_2 of the model in Fig. 1 satisfy the conditions of Def. 3.1 relative to X_1, X_2 and Y. Thus, from the result above, the parameters c_1 and c_2 are identified.

Brito and Pearl have generalized the simple instrumental sets through the use of conditioning.

Definition 3.2 (Brito and Pearl (2002a); Brito (2004))**.** *The set* $\mathbf{Z}$ *is said to be a* generalized instrumental set *relative to* $\mathbf{X}$ *and* Y *in* $\mathcal{G}$ *if for a permutation* $Z_1, \ldots, Z_k$ *of* $\mathbf{Z}$ *and a permutation* $X_1, \ldots, X_k$ *of* $\mathbf{X}$ *there exist triples* $(Z_1, \mathbf{W}_1, \pi_1), \ldots, (Z_k, \mathbf{W}_k, \pi_k)$*, with* $\mathbf{W}_i \subseteq \mathbf{V}$*, such that:*

(a) Every π_i *is an unblocked path between* Z_i *and* Y *including edge* $X_i \to Y$ *and for* $i = 1, \ldots, k$*,* Z_i *and the elements of* $\mathbf{W}_i$ *are non-descendents of* Y*.*
(b) Every set $\mathbf{W}_i$ *d-separates* Z_i *from* Y *in* $\overline{\mathcal{G}}$*; but* $\mathbf{W}_i$ *does not block path* π_i*.*
(c) Paths $\pi_1, \ldots, \pi_k$ *are incompatible.*

Analogously, they prove that if $\{Z_1, \ldots, Z_k\}$ is a generalized instrumental set relative to $\{X_1, \ldots, X_k\}$ and Y then the parameters of edges $X_i \to Y$ can be computed by solving a system of linear equations which involve partial correlations of Z_i and Y given $\mathbf{W}_i$.

Note that by restricting the cardinality k to $k = 1$ for the simple instrumental sets (Def. 3.1) we get just the IV. However, restricting k to 1 in Def. 3.2 leads to a

[1] except for parameterizations $\Theta \in \mathbb{R}^h$ that reside on a subset of Lebesgue measure zero of $\mathbb{R}^h$, where h is the total number of parameters.

new notion of singular conditional instrumental sets which, in general, does not coincide with the concept of conditional instruments. This is because Def. 3.2 requires that X and Z must be connected by a path that is neither blocked by the empty set nor by $\mathbf{W}$, while a conditional instrument only assumes the connection with a path not blocked by $\mathbf{W}$. Actually the case is a further restriction of an "ancestral instrument" (van der Zander et al., 2015). We discuss this case separately in the appendix.

In this paper we introduce a natural intermediate level between the simple and the generalized instrumental sets by restricting Def. 3.2 such that the sets $\mathbf{W}_1 = \mathbf{W}_2 = \ldots = \mathbf{W}_k$ have to be equal.

Definition 3.3. *The set* $\mathbf{Z}$ *is said to be a* simple conditional instrumental set *relative to* $\mathbf{X}$ *and* Y *in* $\mathcal{G}$ *if for a permutation* $Z_1, \ldots, Z_k$ *of* $\mathbf{Z}$ *and a permutation* $X_1, \ldots, X_k$ *of* $\mathbf{X}$ *there exists a set* $\mathbf{W} \subseteq \mathbf{V}$ *and pairs* $(Z_1, \pi_1), \ldots, (Z_k, \pi_k)$*, such that:*

(a) Every π_i *is an unblocked path between* Z_i *and* Y *including edge* $X_i \to Y$ *and all* Z_i *and all elements of* $\mathbf{W}$ *are non-descendents of* Y*.*
(b) $\mathbf{W}$ *d-separates every* Z_i *from* Y *in* $\overline{\mathcal{G}}$*; but* $\mathbf{W}$ *does not block any path* π_i*.*
(c) Paths $\pi_1, \ldots, \pi_k$ *are incompatible.*

As we will show, this definition provides a substantial subclass of generalized instrumental sets (Def. 3.2) that can be verified by an algorithm in polynomial time. The NP-hardness result says that no such algorithm exists for the generalized instrumental sets, unless P = NP.

4 Finding and Testing Instruments

One of the major drawbacks of the IV methods for identification of multiple parameters is that any direct approach to find generalized instrumental sets requires large computational efforts. So far it was not clear, whether generalized instrumental sets, respectively maximal instrumental sets, can be found efficiently (for a more discussion see e.g. (Tian, 2007) or (Brito and Pearl, 2006)). Moreover, until now no results have been known which would demonstrate the intractability of this problem. In our paper we provide a complete answer to these questions.

Assume $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a causal graph of n nodes and m edges. Let Y be a node and let $\mathbf{X}$ be direct causes of Y in $\mathcal{G}$. Our first result shows that the simple instrumental sets can be found by an algorithm running in polynomial time $\mathcal{O}(nm)$.

Theorem 4.1. *There exists an algorithm which for given node* Y *and a set of nodes* $\mathbf{X}$ *in a causal graph* $\mathcal{G}$*, finds simple instrumental sets* $\mathbf{Z}$ *relative to* $(\mathbf{X}, Y)$ *(Def. 3.1), if such a set exists; Otherwise it returns* $\perp$*. The running time of this algorithm is* $\mathcal{O}(nm)$*.*

Importantly, this algorithm is easily implementable and it can be used to find a maximal set of simple instruments, i.e. a set of variables $\mathbf{Z}'$ of maximum cardinality which satisfies conditions of Def. 3.1 relative to $\mathbf{X}' \subseteq \mathbf{X}$ and Y.

Our next result shows that testing whether, for a given $\mathbf{Z}$, there exists a common set $\mathbf{W}$ which satisfies the conditions of generalized instrumental sets can be solved in polynomial time.

Theorem 4.2. *There exists an algorithm which for given node* Y *and node sets* $\mathbf{X}$ *and* $\mathbf{Z}$ *in a causal graph* $\mathcal{G}$*, tests whether* $\mathbf{Z}$ *is a simple conditional instrumental set relative to* $(\mathbf{X}, Y)$ *(Def. 3.3).*

Also this algorithm is easily implementable. Moreover, in cases when k is bounded by a constant, say d, we can use this algorithm to find a simple conditional instrumental set in time $\mathcal{O}(n^{d+3})$. Finding a generalized instrumental set seems to be harder. In fact, in Section 6 we confirm this intuition by proving that testing if a given set $\mathbf{Z}$ is a generalized instrumental set relative to $\mathbf{X}$ and Y is *NP*-complete. However, if k is bounded by a constant, generalized instrumental sets can be found in polynomial time.

Theorem 4.3. *There exists an algorithm which for a given node* Y *and node set* $\mathbf{X}$ *of size* k *in a causal graph* $\mathcal{G}$*, finds a generalized instrumental set* $\mathbf{Z}$ *relative to* $(\mathbf{X}, Y)$ *(Def. 3.2). The running time of this algorithm is* $\mathcal{O}(k(k!)^2 n^{4k+1})$*.*

In the next section we describe the algorithms for Theorems 4.1, 4.2, and 4.3.

5 Polynomial Time Algorithms

For simplicity of presentation we will assume in this section that the causal graph is a DAG having only directed edges, but no bidirected ones. To apply our algorithms for graphs with bidirected edges, for every edge $V_i \leftrightarrow V_j$ we introduce a unique node U, replace $V_i \leftrightarrow V_j$ with $V_i \leftarrow U \rightarrow V_j$ and assume U is an unobservable variable.

So, let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a DAG and let $\mathbf{M} \subseteq \mathbf{V}$ denote the subset of measurable nodes. Moreover, let $n = |\mathbf{V}|$ and $m = |\mathbf{E}|$. We will generally assume that $\mathbf{W} \subseteq \mathbf{M}$, if $\mathbf{W}$ is used for d-separation.

5.1 Nearest Separators

Testing if a certain set is a generalized instrumental set requires one to solve two different problems: finding

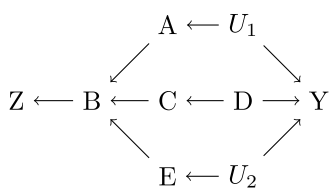

Figure 3: A DAG with 2 unobservable variables $\{U_1, U_2\}$. The only nearest separator of Y and Z is $\{A, D, E\}$.

separating sets $\mathbf{W}_i$ and finding paths π_i. As the $\mathbf{W}_i$ we can use nearest separators, which are defined in (van der Zander et al., 2015) as follows (see Fig. 3 for an example):

Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a graph and let $\mathbf{M} \subseteq \mathbf{V}$ denote the measured nodes and let Y and Z be nodes in $\mathbf{V}$. We say that Y and Z are *separable* in $\mathcal{G}$ if there exists $\mathbf{W} \subseteq \mathbf{M}$ such that $(Z \perp\!\!\!\perp Y \mid \mathbf{W})_{\mathcal{G}}$. For given nodes Y and Z in $\mathbf{V}$ we call a subset $\mathbf{W} \subseteq \mathbf{M} \cap An(Y, Z)$ a *nearest separator*[2] according to (Y, Z) if and only if (i) $(Z \perp\!\!\!\perp Y \mid \mathbf{W})_{\mathcal{G}}$ and (ii) for all $X \in An(Y \cup Z) \setminus \{Y, Z\}$ and any path π in the moral graph $(\mathcal{G}_{An(Y \cup Z)})^m$ connecting X and Z, if there exists $\mathbf{W}' \subseteq \mathbf{M}$ such that $(Z \perp\!\!\!\perp Y \mid \mathbf{W}')_{\mathcal{G}}$ and $\mathbf{W}'$ does not contain a node of π then $\mathbf{W}$ does not contain a node of π either.

They describe an efficient, greedy algorithm to find such a nearest separator:

Lemma 5.1. *(van der Zander et al., 2015) There exists an algorithm that finds a nearest separator $\mathbf{W} \subseteq An(Y \cup Z)$ if Y and Z are separable in $\mathcal{G}$; otherwise it returns $\perp$. Moreover, if Y and Z can be separated in $\mathcal{G}$ by a set that does not contain a descendant of Y, then $\mathbf{W} \subseteq An(Y \cup Z) \setminus De(Y)$. The runtime of the algorithm is $\mathcal{O}(nm)$.*

In the appendix we show that a nearest separator $\mathbf{W}_i$ does not block the π_i of a generalized instrumental set, because it could only block the part of π_i contained in the moral graph, which it does not block by definition, leading to:

Lemma 5.2. *Let Y and Z be nodes in a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, $\mathbf{X}$ a subset of parents of Y, $X \in \mathbf{X}$ a certain node, π an active path between Z and Y including edge $X \to Y$ in $\mathcal{G}$, $\overline{\mathcal{G}} = (\mathbf{V}, \mathbf{E} \setminus (\mathbf{X} \to Y))$, and $\mathbf{W}$ a nearest separator between Y and Z in $\overline{\mathcal{G}}$. If there exists a set $\mathbf{W}'$ that d-separates Y and Z and does not contain a node of π, then $\mathbf{W}$ also does not contain a node of π.*

[2] The definition of the nearest separator used in this paper is stricter than the one given by van der Zander et al. (2015), but their proofs are also valid for our definition.

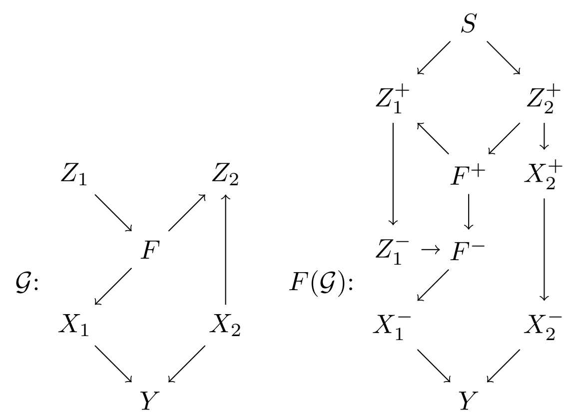

Figure 4: A DAG $\mathcal{G}$ and its flow graph $F(\mathcal{G})$.

5.2 Testing Simple Conditional Instruments

For given $\mathbf{X}$, Y and $\mathbf{Z}$ the separator $\mathbf{W}$ for a simple conditional instrumental set can be computed as the nearest separator according to (Y, Z') in the graph obtained from $\mathcal{G}$ by deleting the edges from $\mathbf{X}$ to Y and adding a new node Z' and edges $Z' \leftarrow Z$ for all $Z \in \mathbf{Z}$.

To find the corresponding paths we transform the graph $\mathcal{G}$ to a *flow graph*, referred as $F(\mathcal{G})$, with respect to $\mathbf{Z}, \mathbf{X}$, and Y. In $F(\mathcal{G})$ collider-free d-paths (treks) become directed paths. Nodes of $F(\mathcal{G})$ consists of two sets, which we denote as $\mathbf{V}^+$ and $\mathbf{V}^-$, as well as Y and a new start node S. The first set, also called $(+)$-layer, is the induced subgraph of all ancestors of $\mathbf{Z}$ with inverted edges. The second set, called $(-)$-layer, the induced subgraph of all ancestors of $\mathbf{X}$. If the same node exists in both layers, the two version of it are distinct but connected by an edge from the $(+)$-layer to the $(-)$-layer. Thus a d-path e.g. containing a fork, becomes a directed path to the fork in the first layer and a directed path from the fork to Y in the second layer[3].

The flow graph $F(\mathcal{G})$ with respect to $\mathbf{Z}, \mathbf{X}, Y$ is formally defined as follows. Let $\mathbf{V}^+ = \{V^+ \mid V \in An(\mathbf{Z})\}$ and $\mathbf{V}^- = \{V^- | V \in An(\mathbf{X})\}$. Then

$$\begin{aligned} V(F(\mathcal{G})) =& \mathbf{V}^+ \cup \mathbf{V}^- \cup \{S, Y\} \\ E(F(\mathcal{G})) =& \{V^+ \to W^+ \mid V, W \in \mathbf{V}^+; V \leftarrow W \in \mathbf{E}\} \cup \\ & \{V^+ \to V^- \mid V \in \mathbf{V}^+ \cap \mathbf{V}^-\} \cup \\ & \{V^- \to W^- \mid V, W \in \mathbf{V}^-; V \to W \in \mathbf{E}\} \cup \\ & \{S \to Z^+ \mid Z \in \mathbf{Z}\} \cup \{X^- \to Y \mid X^- \in \mathbf{X}\}. \end{aligned}$$

In $F(\mathcal{G})$ the disjoint paths π_i correspond to a $|\mathbf{Z}|$-flow from S to Y and can be found with a standard max-flow algorithm with vertex-capacities. S and Y have

[3] Signs $+$ and $-$ can be seen as the arrow head of the edge leaving a node of this layer.

infinite capacity, the $\pm$ nodes resulting from nodes in $\mathbf{W}$ have zero capacity, and all other nodes have unit capacity.

```
function TEST-SIMPLE-COND-IVS(G, X, Y, Z)
    Construct graph G' from G by:
        adding a node Z', edges Z' ← Z, and
        removing all edges X → Y from G
    Let W be a nearest separator for (Y, Z') in G'
    if W = ⊥ ∨ W ∩ De(Y) ≠ ∅ ∨ Z ∩ W ≠ ∅ then
        return false
    Construct F(G) with respect to Z, X, and Y
    Assign capacities to the nodes of F(G):
        infinite capacity to S, Y,
        zero capacity to nodes stemming from W,
        unit capacity to all other nodes
    if a |Z|-flow from S to Y exists in G' then
        return true else return false
```

Figure 5: Test-Simple-Cond-IVs

The complete algorithm to test simple conditional instrumental sets is given in Fig. 5. For a proof that it satisfies the requirements of Theorem 4.2. see the appendix.

5.3 Finding Simple Instruments

In this section we duscuss an algorithm to find a simple instrumental set (Def. 3.1). Testing if a given $\mathbf{Z}$ fulfills the conditions of simple instruments can be done in time $\mathcal{O}(nm)$ using the algorithm TEST-SIMPLECOND-IVS presented in Section 5.2. To this aim we modify the algorithm by replacing the calculation of $\mathbf{W}$ as nearest separator with a fixed $\mathbf{W} = \emptyset$.

The algorithm (see Fig. **??** in the appendix) which satisfies the requirements of Theorem 4.1 is a modification of algorithm TEST-SIMPLECOND-IVS. Basically, instead finding a maximum flow from S to Y through $\mathbf{Z}$, we search a flow from S to Y through every node in $De(An(\mathbf{X}))$ that might be in $\mathbf{Z}$. The proof of its correctness can be found in the appendix.

5.4 Testing Generalized Instruments

Generalized vertex disjoint paths problem. Let us now reconsider the problem of finding the paths π_i in the general case. In this case the endnodes of the paths are not interchangeable, so it cannot be solved with a network flow. However, for a fixed k, finding k paths that are just node-disjoint is a well-researched problem (k-vertex disjoint paths problem, k-VDPP or k-linkage), and known to be NP-complete in general directed graphs (Garey and Johnson, 1979) but solvable in DAGs in polynomial time (Fortune et al., 1980).

The problem k-VDPP asks, given $2k$ not necessarily distinct nodes $(s_1, \dots, s_k), (t_1, \dots, t_k)$ if there are k paths from each s_i to t_i that do not share a common node except for the end nodes.

We generalize k-VDPP to find directed paths that satisfy the following conditions:

Definition 5.3 (Generalized vertex disjoint paths problem (k-GVDPP))**.** *Let $(S_1, \dots, S_k), (T_1, \dots, T_k)$ be $2k$ not necessarily distinct nodes of a DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, let $\mathbf{W}_1, \dots, \mathbf{W}_k \subseteq \mathbf{V}$ be sets of nodes, with $S_i, T_i \notin \mathbf{W}_i$, and let $\mathbf{C} \subseteq \{\{i, j\} \mid 1 \leq i, j \leq k\}$ be a set of pairs. Question: Do there exist paths p_i, s.t.*

1. *p_i is a directed path from S_i to T_i,*
2. *p_i does not contain a node of $\mathbf{W}_i$, and*
3. *p_i does not share a node with p_j, $i \neq j$, unless that node is S_i, T_i, S_j, T_j; or $\{i, j\} \in \mathbf{C}$.*

We generalize the pebbling game algorithm given by Perl and Shiloach (1978) for $k = 2$ and generalized by Fortune et al. (1980) to arbitrary k.

Our pebble game is defined by the following rules of which rule 2 and 3 may be applied arbitrary often in any order. In the description below, the level of a node V is defined to be the length of the longest, directed path starting at V.

1. Initially: use k pebbles p_i and place p_i on S_i.
2. Pebble p_i may be moved along a directed edge $V \to W$ if W is not in $\mathbf{W}_i$ and
   - V has the largest level of any pebbled node and
   - there is no pebble p_j on W unless $\{i, j\} \in \mathbf{C}$ or $W \in \{S_j, T_i\}$.
3. Pebble p_i may be removed once it reaches T_i.
4. The game is won if all pebbles are removed.

In the appendix we prove that this game is equivalent to the k-GVDPP and that it can be played efficiently:

Lemma 5.4. *The pebbling game can be won iff there exists a solution to k-GVDPP.*

Lemma 5.5. *There exists an $\mathcal{O}(k(n+1)^{k+1})$ algorithm to solve k-GVDPP.*

Reducing instrumental set testing. We show that a test if a given instance is a generalized instrumental set can be done by an algorithm which has access to a subroutine for solving k-GVDPP. The algorithm begins by creating a nearest separator according to (Y, Z_i) in $\overline{\mathcal{G}}$ for each i to use it as set $\mathbf{W}_i$. Next it enumerates all permutations $Z_1, \dots, Z_k$ of $\mathbf{Z}$ and $X_1, \dots, X_k$ of $\mathbf{X}$ as well as all combinations for directed and fork paths for each π_i, i.e. π_i is considered either as directed from Z_i to X_i, directed from X_i to Z_i, or containing a fork F_i. Knowing the direction and/or fork of a d-path, we can treat it as one or two directed paths. From condition *(c)* of Def. 3.2 it follows that two of these directed paths can only intersect each other iff one path is directed towards a Z_i and the other path towards an X_j with $i < j$. These nodes and constraints directly correspond to a k-GVDPP instance with up to $2k$ nodes. If one of these k-GVDPP instances has a solution, $\mathbf{Z}$ is a generalized instrumental set[4]. The details of this algorithm can be found in the appendix. Thus, we obtain the following:

Lemma 5.6. *There exists an algorithm which for a given Y and sets of k nodes $\mathbf{X}$ and $\mathbf{Z}$, using a solver for GVDPP tests if $\mathbf{Z}$ is a generalized instrumental set relative to $\mathbf{X}$ and Y calling the solver $\mathcal{O}((k!)^2 n^k)$ times for k'-GVDPP instances, with $k' \in \{k, \ldots, 2k\}$.*

Corollary 5.7. *Given Y and sets $\mathbf{X}, \mathbf{Z}$ containing k nodes, we can test if $\mathbf{Z}$ is a generalized instrumental set relative to $\mathbf{X}$ and Y in time $\mathcal{O}(k(k!)^2 n^{3k+1})$.*

This corollary implies Theorem 4.3.

6 Intractability Result

Now we prove that it is an NP-complete problem to test if a given set is a generalized instrumental set:

Theorem 6.1. *Given a DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, a node Y and sets $\mathbf{X}, \mathbf{Z} \subset \mathbf{V}$ determining if $\mathbf{Z}$ is a generalized instrumental set relative to $\mathbf{X}$ and Y (Def. 3.2) is an NP-complete problem.*

Proof. Obviously the conditions of Def. 3.2 can be easily verified after guessing the tuples. Thus, the problem is in NP. To prove the NP-hardness, we show a polynomial time reduction from 3-SAT to the problem. Let $\mathbf{V}$ be a set of n_V variables and let $\mathcal{C} = (V_{1,1} \vee V_{1,2} \vee V_{1,3}) \wedge (V_{2,1} \vee V_{2,2} \vee V_{2,3}) \wedge \ldots (V_{n_C,1} \vee V_{n_C,2} \vee V_{n_C,3})$ with $V_{i,j} \in \mathbf{V} \cup \{\overline{V} \mid V \in \mathbf{V}\}$ be a 3-SAT instance with n_C clauses. The variables $\mathbf{V} = \{V_i \mid 1 \le i \le n_V\}$ and clauses of $\mathcal{C} = \{C_i \mid 1 \le i \le n_C\}$ are arbitrarily indexed. Let $o_i = |\{C \in \mathcal{C} \mid V_i \in C\}|$, resp. $\overline{o}_i$, denote the number of occurrences of literal V_i, resp. $\overline{V}_i$, in $\mathcal{C}$. W.l.o.g. we assume $o_i > 0$ and $\overline{o}_i > 0$.

We adapt the proof given by Even et al. (1976) for multi-commodity flows to instrumental sets. So we construct a DAG $\mathcal{G}$ as shown in Fig. 6.

$\mathcal{G}$ has the following nodes:

$$
\begin{aligned}
\mathbf{V}_{\mathcal{G}} = \ & \{Y, Z_0', Z_0, \ldots, Z_{n_C}, X_0, \ldots, X_{n_C}\} \\
& \cup \{C_1, \ldots, C_{n_C}, D_1, \ldots, D_{n_C}\} \\
& \cup \{V_i^s, V_i^t \mid 1 \le i \le n_V\} \\
& \cup \{V_i^j \mid 1 \le i \le n_V \wedge 1 \le j \le o_i\} \\
& \cup \{\overline{V}_i^j \mid 1 \le i \le n_V \wedge 1 \le j \le \overline{o}_i\}
\end{aligned}
$$

[4] The algorithm also has to consider various, cumbersome cases of endnodes in $\mathbf{Z} \cup \mathbf{X}$ that might occur in other paths.

and edges:

$$
\begin{aligned}
\mathbf{E} = \ & \{Z_0 \leftarrow Z_0' \rightarrow V_1^s\} \\
& \cup \{V_i^s \rightarrow V_i^1 \rightarrow \ldots \rightarrow V_i^{o_i} \rightarrow V_i^t \mid 1 \le i \le n_V\} \\
& \cup \{V_i^s \rightarrow \overline{V}_i^1 \rightarrow \ldots \rightarrow \overline{V}_i^{\overline{o}_i} \rightarrow V_i^t \mid 1 \le i \le n_V\} \\
& \cup \{V_i^t \rightarrow V_{i+1}^s \mid 1 \le i \le n_V - 1\} \\
& \cup \{V_{n_V}^t \rightarrow X_0 \rightarrow Y\} \\
& \cup \{Z_i \rightarrow V_j^k \mid 1 \le i \le n_C \wedge 1 \le j \le n_V \wedge 1 \le k \le o_j\} \\
& \cup \{Z_i \rightarrow \overline{V}_j^k \mid 1 \le i \le n_C \wedge 1 \le j \le n_V \wedge 1 \le k \le \overline{o}_j\} \\
& \cup \{V_j^k \rightarrow C_i \mid \text{ the } k\text{-th occurrence of } V_j \text{ is in } C_i\} \\
& \cup \{\overline{V}_j^k \rightarrow C_i \mid \text{ the } k\text{-th occurrence of } \overline{V}_j \text{ is in } C_i\} \\
& \cup \{C_i \rightarrow D_i \rightarrow X_i \rightarrow Y \mid 1 \le i \le n_V\} \\
& \cup \{Y \leftrightarrow D_i \rightarrow Z_0 \mid 1 \le i \le n_V\}
\end{aligned}
$$

We use indices 0 to n_C for X_i instead of 1 to $n_C + 1$ to simplify the notation. We claim that there exists an assignment to $V_1, \ldots, V_{n_V}$ that satisfies $\mathcal{C} = \bigwedge_i C_i$ iff $\mathbf{Z} = \{Z_0, \ldots, Z_{n_C}\}$ is a generalized instrumental set relative to $\mathbf{X} = \{X_0, \ldots, X_{n_C}\}$ and Y in $\mathcal{G}$.

"$\Leftarrow$": Assume $\mathbf{Z}$ is a generalized instrumental set. Then there exist tuples $(Z_{i_0}, \mathbf{W}_0, \pi_0)$, $(Z_{i_1}, \mathbf{W}_1, \pi_1), \ldots, (Z_{i_{n_C}}, \mathbf{W}_{n_C}, \pi_{n_C})$ satisfying Def. 3.2. First we show that the path from Z_0 actually ends at $X_0 \rightarrow Y$. There are active paths $Y \leftrightarrow D_i \rightarrow Z_0$ for all D_i, which need to be blocked. Thus nodes D_i are in the $\mathbf{W}_j$ associated with the path starting at Z_0, so the path cannot contain D_i. Since $X_1, \ldots, X_{n_C}$ can only be reached by traversing $D_1, \ldots, D_{n_C}$, the path has to end at X_0.

Since the nodes $Z_1, \ldots, Z_{n_C}$ are all connected to exactly the same nodes, we can assume w.l.o.g. that path π_i starts at Z_i.

Every node V_i^j is only visited by a directed subpath $\rightarrow V_i^j \rightarrow$ because every path can only enter it through a $\rightarrow$ edge. So none of these nodes is visited by two paths. Otherwise condition *(c)* of Def. 3.2 (that the subpath $\pi_{i'}[V_i^j \sim X_{i'}]$ has to point to V_i^j) would be violated.

Since path π_0 can neither visit node C_i nor Z_i for $i > 0$ through a collider, it visits V_i^s and then passes either through the upper path or the lower path to V_i^t. We assign the following values to the variables V_i

$$
V_i := \begin{cases} \text{true} & \text{if } V_i^1 \notin \pi_0, \\ \text{false} & \text{otherwise.} \end{cases}
$$

This assignment satisfies the formula: Assume there is clause C_i that is not satisfied. We know that path π_i has the form $Z_i \rightarrow W_k^j \rightarrow \ldots W_k^{j'} \rightarrow C_i \rightarrow D_i \rightarrow X_i \rightarrow Y$ for W's corresponding to one variable V_k or its negation $\overline{V}_k$, since π_i cannot cross through V_k^t to

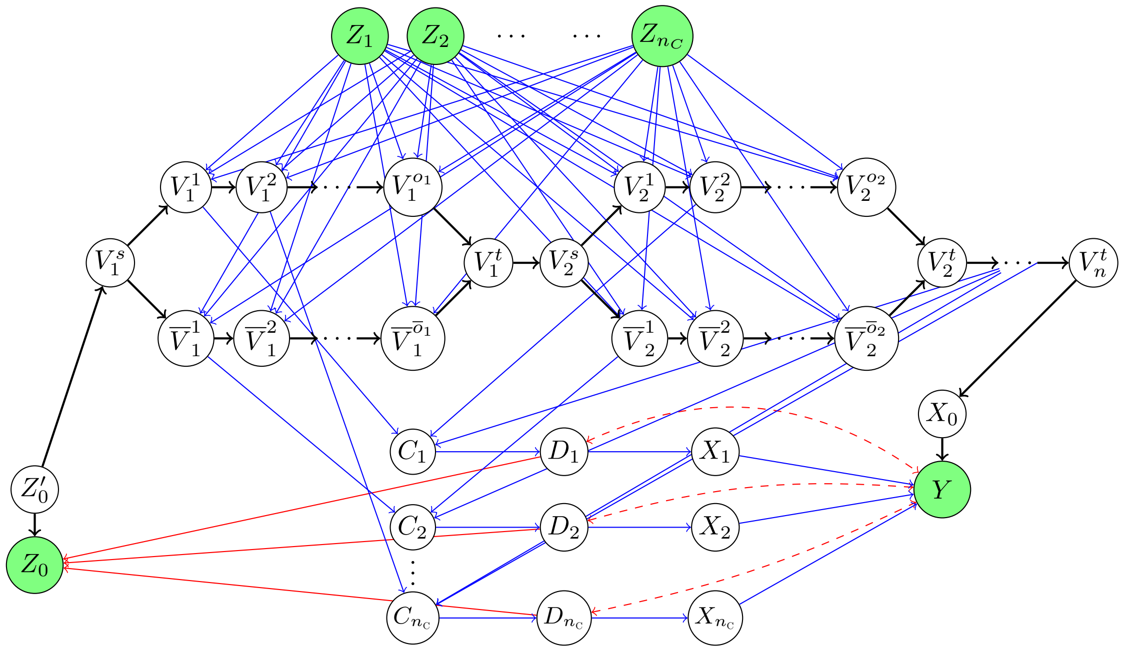

Figure 6: A graph $\mathcal{G}$ with a generalized instrumental set $\mathbf{Z}$ constructed from a 3-SAT instance.

another lobe; Otherwise it would intersect π_0 at V_k^t. Also $W_k^t \notin \pi_0$. If W_k^j corresponds to V_k then V_k is true and clause C_i contains variable V_k. If W_k^j corresponds to $\overline{V}_k$, V_k is false and C_i contains the negation. So C_i is satisfied.

"$\Rightarrow$": Let $V_i \in \{\text{true}, \text{false}\}$ be a satisfying assignment for the variables V_i. Assume C_i is satisfied by a literal $W \in C_i$ which is the k-th occurrence of a variable V_j in $\mathcal{C}$. Let $v(C_i) \in \{V_j^k, \overline{V}_j^k\}$ be the node corresponding to W. Let $p(i) = V_i^1 \to \ldots \to V_i^{o_i}$ if $V_i = \textit{false}$; Otherwise let $p(i) = \overline{V}_i^1 \to \ldots \to \overline{V}_i^{\overline{o}_i}$. We choose the following tuples which satisfy the conditions of Def. 3.2:

- $(Z_0, \{C_i, D_i \mid 1 \le i \le n_C\}$, $Z_0 \leftarrow Z_0' \to V_1^s \to p(1) \to V_1^t \to V_2^s \to p(2) \to V_2^t \to V_3^s \to \ldots \to p(n_V) \to V_{n_V}^t \to X_0 \to Y)$,
- $(Z_1, \emptyset, Z_1 \to v(C_1) \to C_1 \to D_1 \to X_1 \to Y)$,
- $\ldots$,
- $(Z_{n_C}, \emptyset, Z_{n_C} \to v(C_{n_C}) \to C_{n_C} \to D_{n_C} \to X_{n_C} \to Y)$.

(a) Y does not have any descendants and any π_i is an unblocked path connecting Z_i with $X_i \to Y$.

(b) In $\overline{\mathcal{G}}$ all paths starting at Y begin with $Y \leftrightarrow D_i$. In the first tuple the paths $Y \leftrightarrow D_i \to Z_0$ are blocked by D_i and the paths $Y \leftrightarrow D_i \leftarrow C_i \leftarrow$ are blocked by C_i. In all other tuples the paths $Y \leftrightarrow D_i \to Z_0$ are irrelevant and the paths $Y \leftrightarrow D_i \leftarrow C_i \leftarrow$ are blocked by D_i. No path π_i is blocked by $\mathbf{W}_i$.

(c) No path $\pi_1, \ldots, \pi_{n_C}$ has a common node with π_0. Otherwise a node V_k^j would correspond to a variable V_k that is false, but literal V_k satisfies clause C_i; or a variable that is true but literal $\overline{V}_k$ satisfies C_i. Paths $\pi_1, \ldots, \pi_{n_C}$ are vertex disjoint or the k-th occurrence of a variable would be in two different clauses. $\square$

7 Conclusions and Future Work

In the paper we have shown that testing, if a given set is a generalized instrumental set, is an NP-complete problem, but it can be solved with a polynomial time algorithm under the assumption of a constant set size.

We give a practically implementable $\mathcal{O}(nm)$ algorithm for special cases, in which the connections between $\mathbf{Z}$ and $\mathbf{X}$ are arbitrary, i.e. every Z_i can be connected to any X_j. The hardness arises in the case when Z_i has to be matched to X_i (or even just Z_1 to X_1, while the remaining connections are arbitrary), which is a little surprising, since one could assume that knowing and verifying a matching would be easier than finding one.

We also give an $\mathcal{O}(nm)$ algorithm to directly find a simple instrumental set. It is an open problem, if the more general cases of instrumental sets can be found without enumerating all possible sets. An interesting problem for future research is also to find an efficiently testable subclass of generalized instruments which is larger than the simple conditional instrumental sets provided in this paper.

Acknowledgements

This work was supported by DFG grant LI 634/4-1.

References

K. A. Bollen. *Structural equations with latent variables.* John Wiley & Sons, 1989.

R. Bowden and D. Turkington. *Instrumental variables.* Cambridge University Press, 1984.

C. Brito. *Graphical Methods for Identification in Structural Equation Models.* PhD Thesis, Dept. of Comp. Sc., University of California, Los Angeles, 2004.

C. Brito. Instrumental sets. In R. Dechter, H. Geffner, and J. Y. Halpern, editors, *Heuristics, Probability and Causality. A Tribute to Judea Pearl*, chapter 17, pages 295–308. College Publications, 2010.

C. Brito and J. Pearl. Generalized instrumental variables. In *Proc. UAI*, pages 85–93, 2002a.

C. Brito and J. Pearl. A graphical criterion for the identification of causal effects in linear models. In *Proc. AAAI*, pages 533–538, 2002b.

C. Brito and J. Pearl. Graphical condition for identification in recursive SEM. In *Proc. UAI*, pages 47–54, 2006.

B. Chen, J. Tian, and J. Pearl. Testable implications of linear structural equation models. In *Proc. AAAI*, pages 2424–2430, 2014.

O. D. Duncan. *Introduction to structural equation models.* Academic Press, 1975.

S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4):691–703, 1976.

F. M. Fisher. *The identification problem in econometrics.* McGraw-Hill, 1966.

S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111 – 121, 1980.

M. Garey and D. Johnson. *Computers and intractability: a guide to the theory of NP-completeness.* WH Freeman & Co., 1979.

J. Pearl. *Causality.* Cambridge University Press, 2009.

Y. Perl and Y. Shiloach. Finding two disjoint paths between two pairs of vertices in a graph. *J. ACM*, 25(1):1–9, 1978.

J. Tian. A criterion for parameter identification in structural equation models. In *Proc. UAI*, pages 392–399, 2007.

B. van der Zander, J. Textor, and M. Liśkiewicz. Efficiently finding conditional instruments for causal inference. In *Proc. IJCAI*, pages 3243–3249, 2015.

S. Wright. The method of path coefficients. *The Annals of Mathematical Statistics*, 5(3):161–215, 1934.