On Searching for Generalized Instrumental Variables

Benito van der Zander Maciej Liškiewicz
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-
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 (S), and sometimes abbreviate singleton sets as $S = \{s\}$. Graphs are written calligraphically (G), and variables in upper-case (X).

We consider graphs $G = (V, E)$ with nodes (vertices, variables) $V$ and directed (A $\rightarrow$ B) and bidirected (A $\leftrightarrow$ B) edges $E$. Nodes linked by an edge are adjacent. A path of length $\ell$ 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 $Y$ we denote by $An(Y)$ the set of all ancestors of nodes in $Y$. For a path $\pi$ we denote by $\pi[A_i \sim A_j]$ the subpath of $\pi$ consisting of the nodes $A_i, A_{i+1}, \ldots, A_j$.

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

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 $\pi_i$; and, if paths $\pi_i$ and $\pi_j$ 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

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

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 $\varepsilon_j$ 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 $\Psi$.

The structure of a linear model over $V_1, \ldots, V_n$ can be represented by a DAG $G$, called a causal graph, whose nodes $V$ correspond to the model’s variables and edges indicate the non-zero parameters of the model; $G$ contains a directed edge $V_i \rightarrow V_j$ if $V_i$ appears in Eq. (1) on the right hand side of the equation for $V_j$ with $c_{ji} \neq 0$ and $G$ 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.

![Figure 1: The linear model and its causal graph.](image)

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

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

The identification problem consists of recovering the parameters $C$ given the observed covariance matrix $\Sigma$ 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 $\Sigma$ with unknowns $C$ which is independent of the unobserved error covariance $\Psi$.

If, given $\Sigma$ and a causal graph, there exists a unique solution $c_{ji}$ satisfying Eq. (2), independent of $\Psi$, 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.
3 Identification with IV Methods

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

However, the graphical characterization of conditional instruments does not indicate how to find \( Z \) and \( W \). 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 \( Y = c_1 X_1 + \ldots + c_k X_k + \varepsilon \), 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 \( X_1 \rightarrow Y, \ldots, X_k \rightarrow Y \) be edges representing directed causes of \( Y \) in the causal diagram \( G = (V,E) \) of a linear model. Let \( \overline{G} \) be the graph obtained from \( G \) by deleting edges \( X_i \rightarrow Y, \ldots, X_k \rightarrow Y \) from \( G \). 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 \( Z \) is said to be a simple instrumental set relative to \( X \) and \( Y \) in \( G \) if for a permutation \( Z_1, \ldots, Z_k \) of \( Z \) and a permutation \( X_1, \ldots, X_k \) of \( X \) 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 \( \overline{G} \).

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:

\[
\rho_{Z_1,Y} = a_{11} c_1 + \ldots + a_{1k} c_k \\
\vdots \\
\rho_{Z_k,Y} = a_{k1} c_1 + \ldots + a_{kk} c_k,
\]

where \( a_{ij} = \rho_{Z_i,Y} \) 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 \( Z \) is said to be a generalized instrumental set relative to \( X \) and \( Y \) in \( G \) if for a permutation \( Z_1, \ldots, Z_k \) of \( Z \) and a permutation \( X_1, \ldots, X_k \) of \( X \) there exist triples \( (Z_i, W_i, \pi_i), \ldots, (Z_k, W_k, \pi_k) \), with \( W_i \subseteq V \), such that:

(a) Every \( \pi_i \) is an unblocked path between \( Z_i \) and \( Y \) including edge \( X_i \rightarrow Y \) and for \( i = 1, \ldots, k \), \( Z_i \) and the elements of \( W_i \) are non-descendants of \( Y \).

(b) Every set \( W_i \) \( d \)-separates \( Z_i \) from \( Y \) in \( \overline{G} \); but \( W_i \) does not block path \( \pi_i \).

(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 \( X_i \rightarrow Y \) can be computed by solving a system of linear equations which involve partial correlations of \( Z_i \) and \( Y \) given \( W_i \).

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 \(W\), while a conditional instrument only assumes the connection with a path not blocked by \(W\). 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 \(W_1 = W_2 = \ldots = W_k\) have to be equal.

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

(a) Every \(\pi_i\) is an unblocked path between \(Z_i\) and \(Y\) including edge \(X_i \rightarrow Y\) and all \(Z_i\), and all elements of \(W\) are non-descendents of \(Y\).

(b) \(W\) \(d\)-separates every \(Z_i\) from \(Y\) in \(G\); but \(W\) does not block any path \(\pi_i\).

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

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

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 \(Z'\) of maximum cardinality which satisfies conditions of Def. 3.1 relative to \(X' \subseteq X\) and \(Y\).

Our next result shows that testing whether, for a given \(Z\), there exists a common set \(W\) 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 \(X\) and \(Z\) in a causal graph \(G\), tests whether \(Z\) is a simple conditional instrumental set relative to \((X,Y)\) (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 \(O(n^{d+3})\). 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 \(Z\) is a generalized instrumental set relative to \(X\) 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 \(X\) of size \(k\) in a causal graph \(G\), finds a generalized instrumental set \(Z\) relative to \((X,Y)\) (Def. 3.2). The running time of this algorithm is \(O(k(k!)^2n^{k+1})\).

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 \(G = (V, E)\) be a DAG and let \(M \subseteq V\) denote the subset of measurable nodes. Moreover, let \(n = |V|\) and \(m = |E|\). We will generally assume that \(W \subseteq M\), if \(W\) 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 \(W_i\) and finding paths \(\pi_i\). As the \(W_i\) we can use nearest separators, which are defined in (van der Zander et al., 2015) as follows (see Fig. 3 for an example):

Let \(G = (V, E)\) be a graph and let \(M \subseteq V\) denote the measured nodes and let \(Y\) and \(Z\) be nodes in \(V\). We say that \(Y\) and \(Z\) are separable in \(G\) if there exists \(W \subseteq M\) such that \((Z \perp Y \mid W)_G\). For given nodes \(Y\) and \(Z\) in \(V\) we call a subset \(W \subseteq M \cap \text{An}(Y,Z)\) a nearest separator\(^2\) according to \((Y, Z)\) if and only if (i) \((Z \perp Y \mid W)_G\) and (ii) for all \(X \in \text{An}(Y \cup Z) \setminus \{Y, Z\}\) and any path \(\pi\) in the moral graph \((G_{\text{An}(Y \cup Z)})^m\) connecting \(X\) and \(Z\), if there exists \(W' \subseteq M\) such that \((Z \perp Y \mid W')_G\) and \(W'\) does not contain a node of \(\pi\) then \(W\) does not contain a node of \(\pi\) 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 \(W \subseteq An(Y \cup Z)\) if \(Y\) and \(Z\) are separable in \(G\); otherwise it returns \(\perp\). Moreover, if \(Y\) and \(Z\) can be separated in \(G\) by a set that does not contain a descendant of \(Y\), then \(W \subseteq An(Y \cup Z) \setminus De(Y)\). The runtime of the algorithm is \(O(nm)\).

In the appendix we show that a nearest separator \(W_i\) does not block the \(\pi_i\) of a generalized instrumental set, because it could only block the part of \(\pi_i\) 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 \(G = (V, E), X\) a subset of parents of \(Y\), \(X \subseteq V\) a certain node, \(\pi\) an active path between \(Z\) and \(Y\) including edge \(X \rightarrow Y\) in \(G\), \(\overline{G} = (V, E \setminus (X \rightarrow Y))\), and \(W\) a nearest separator between \(Y\) and \(Z\) in \(\overline{G}\). If there exists a set \(W'\) that \(d\)-separates \(Y\) and \(Z\) and does not contain a node of \(\pi\), then \(W\) also does not contain a node of \(\pi\).

\(^2\)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.

\(^3\)Signs + and – can be seen as the arrow head of the edge leaving a node of this layer.

5.2 Testing Simple Conditional Instruments

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

To find the corresponding paths we transform the graph \(G\) to a flow graph, referred as \(F(G)\), with respect to \(Z, X, Y\). In \(F(G)\) collider-free \(d\)-paths (treks) become directed paths. Nodes of \(F(G)\) consists of two sets, which we denote as \(V^+\) and \(V^-\), as well as \(Y\) and a new start node \(S\). The first set, also called \((+)-layer\), is the induced subgraph of all ancestors of \(Z\) with inverted edges. The second set, called \((-)-layer\), the induced subgraph of all ancestors of \(X\). 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 \(F(G)\) with respect to \(Z, X, Y\) is formally defined as follows. Let \(V^+ = \{V^+ \mid V \in \text{An}(Z)\}\) and \(V^- = \{V^- \mid V \in \text{An}(X)\}\). Then

\[ V(F(G)) = V^+ \cup V^- \cup \{S, Y\} \]

\[ E(F(G)) = \{V^+ \rightarrow W^+ \mid V, W \in V^+; V \leftarrow W \in E\} \cup \{V^+ \rightarrow V^- \mid V \in V^+ \cap V^-\} \cup \{V^- \rightarrow W^- \mid V, W \in V^-; V \rightarrow W \in E\} \cup \{S \rightarrow Z^+ \mid Z \in Z\} \cup \{X^- \rightarrow Y \mid X^- \in X\}. \]

In \(F(G)\) the disjoint paths \(\pi_i\) correspond to a \(|Z|\)-flow from \(S\) to \(Y\) and can be found with a standard max-flow algorithm with vertex-capacities. \(S\) and \(Y\) have
infinite capacity, the ± nodes resulting from nodes in \( W \) 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 \( Z' \leftarrow Z \), and
- removing all edges \( X \rightarrow Y \) from \( G \)

Let \( W \) be a nearest separator for \((Y, Z')\) in \( G' \)

if \( W = \perp \lor W \cap De(Y) \neq \emptyset \lor Z \cap W \neq \emptyset \) then return false

Construct \( F(G) \) with respect to \( Z, X, Y \), and \( X \)

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 discuss an algorithm to find a simple instrumental set (Def. 3.1). Testing if a given \( Z \) fulfills the conditions of simple instruments can be done in time \( O(nm) \) using the algorithm Test-SimpleCond-IVs presented in Section 5.2. To this aim we modify the algorithm by replacing the calculation of \( W \) as nearest separator with a fixed \( W = \emptyset \).

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 of finding a maximum flow from \( S \) to \( Y \) through \( Z \), we search a flow from \( S \) to \( Y \) through every node in \( De(An(X)) \) that might be in \( Z \). 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 \( \pi_i \) 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 \((k\text{-vertex disjoint paths problem, } k\text{-VDPP or } k\text{-linkage})\), and known to be \( \text{NP}\)-complete in general directed graphs (Garey and Johnson, 1979) but solvable in DAGs in polynomial time (Fortune et al., 1980).

The problem \( k\text{-VDPP} \) asks, given \( 2k \) not necessarily distinct nodes \((s_1, \ldots, s_k), (t_1, \ldots, t_k) \) 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 \((k\text{-GVDPP})\)). Let \((S_1, \ldots, S_k), (T_1, \ldots, T_k)\) be \( 2k \) not necessarily distinct nodes of a DAG \( G = (V, E) \), let \( W_1, \ldots, W_k \subseteq V \) be sets of nodes, with \( S_i, T_i \notin W_i \), and let \( C \subseteq \{(i, j) \mid 1 \leq i, j \leq k\} \) 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 \( W_i \), and
3. \( p_i \) does not share a node with \( p_j \), \( i \neq j \), unless that node is \( S_i, T_i, S_j, T_j \; \text{or} \; \{i, j\} \in C \).

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 \( V \rightarrow W \) if \( W \) is not in \( W_i \) and
   - \( V \) has the largest level of any pebbled node and
   - there is no pebble \( p_j \) on \( W \) unless \( \{i, j\} \in C \) or \( W \in \{S_j, T_j\} \).
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 \( k\text{-GVDPP} \) and that it can be played efficiently:

**Lemma 5.4.** The pebbling game can be won iff there exists a solution to \( k\text{-GVDPP} \).

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

**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 \( k\text{-GVDPP} \). The algorithm begins by creating a nearest separator according to \((Y, Z)\) in \( G \) for each \( i \) to use it as set \( W_i \). Next it enumerates all permutations \( Z_1, \ldots, Z_k \) of \( Z \) and \( X_1, \ldots, X_k \) of \( X \) as well as all combinations for directed and fork paths for each \( \pi_i \), i.e. \( \pi_i \) 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 $k$-GVDP instance with up to $2k$ nodes. If one of these $k$-GVDP instances has a solution, $Z$ is a generalized instrumental set\textsuperscript{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 $X$ and $Z$, using a solver for GVDP tests if $Z$ is a generalized instrumental set relative to $X$ and $Y$ calling the solver $O((k!)^3 n^k)$ times for $k'$-GVDP instances, with $k' \in \{k, \ldots, 2k\}$.

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

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 $\mathcal{G} = (V, E)$, a node $Y$ and sets $X, Z \subset V$ determining if $Z$ is a generalized instrumental set relative to $X$ 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 $V$ be a set of $n_V$ variables and let $C = (\overline{V}_{1,1} \lor V_{1,2} \lor V_{1,3}) \land (\overline{V}_{2,1} \lor V_{2,2} \lor V_{2,3}) \land \ldots (\overline{V}_{n_C, 1} \lor V_{n_C, 2} \lor V_{n_C, 3})$ with $V_{i,j} \in V \lor \{T \mid V \in V\}$ be a 3-SAT instance with $n_C$ clauses. The variables $V = \{V_i \mid 1 \leq i \leq n_V\}$ and clauses of $C = \{C_i \mid 1 \leq i \leq n_C\}$ are arbitrarily indexed. Let $o_i = |\{C \in C \mid V_i \in C\}|$, resp. $\sigma_i$, denote the number of occurrences of literal $V_i$, resp. $\overline{V}_i$, in $C$. W.l.o.g. we assume $o_i > 0$ and $\sigma_i > 0$.

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:

$V_\mathcal{G} = \{Y, Z_0', Z_0, \ldots, Z_{n_C}, X_0, \ldots, X_{n_C}\}$

- $\{C_1, \ldots, C_{n_C}, D_1, \ldots, D_{n_C}\}$
- $\{V_i^s, V_i^t \mid 1 \leq i \leq n_V\}$
- $\{V_i^s \mid 1 \leq i \leq n_V \land 1 \leq j \leq o_i\}$
- $\{V_i^t \mid 1 \leq i \leq n_V \land 1 \leq j \leq \sigma_i\}$

\[ E = \{Z_0 \leftrightharpoons Z_0' \to V_1^s\} \]

\[ \cup \{V_i^s \to V_i^t \to \ldots \to V_i^{t_0} \to V_i^s \mid 1 \leq i \leq n_V\} \]

\[ \cup \{V_i^t \to V_i^s \mid 1 \leq i \leq n_V - 1\} \]

\[ \cup \{V_i^t, \to X_0 \to Y\} \]

\[ \cup \{Z_i \to V_i^k \mid 1 \leq i \leq n_C \land 1 \leq j \leq n_V \land 1 \leq k \leq o_i\} \]

\[ \cup \{V_j \to V_j^k \mid 1 \leq j \leq n_C \land \sigma_j < 1 \leq \leq n_V \land 1 \leq k \leq \sigma_j\} \]

\[ \cup \{V_j^k \to C_i \mid \text{the k-th occurrence of } V_j \text{ is in } C_i\} \]

\[ \cup \{V_j^k \to C_i \mid \text{the k-th occurrence of } V_j \text{ is in } C_i\} \]

\[ \cup \{C_i \to D_i \to X_i \to Y \mid 1 \leq i \leq n_V\} \]

\[ \cup \{Y \leftrightarrow D_i \to Z_i \mid 1 \leq i \leq n_V\} \]

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 $C = \bigwedge_i C_i$ iff $Z = \{Z_0, \ldots, Z_{n_C}\}$ is a generalized instrumental set relative to $X = \{X_0, \ldots, X_{n_C}\}$ and $Y$ in $\mathcal{G}$.

"$\leq$": Assume $Z$ is a generalized instrumental set. Then there exist tuples $(Z_{i_0}, W_{r_0}, \rho_0), (Z_{i_1}, W_{r_1}, \rho_1), \ldots, (Z_{i_{n_C}}, W_{r_{n_C}}, \rho_{n_C})$ satisfying Def. 3.2. First we show that the path from $Z_0$ actually ends at $X_0$. There are active paths $Y \leftrightarrow D_i \to Z_0$ for all $D_i$, which need to be blocked. Thus nodes $D_i$ are in the $W_j$ 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 $\pi_i$ starts at $Z_i$.

Every node $V_i^s$ is only visited by a directed subpath $\to V_i^t \to$ because every path can only enter it through $\to$ edge. So none of these nodes is visited by two paths. Otherwise condition (c) of Def. 3.2 (that the subpath $\pi_i[V_i^s \sim V_i^t]$ has to point to $V_i^t$) would be violated.

Since path $\pi_0$ 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 $V_i$:

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

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

\textsuperscript{4}The algorithm also has to consider various, cumbersome cases of endnodes in $Z \cup X$ that might occur in other paths.
On Searching for Generalized Instrumental Variables

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 \( \pi_0 \) at \( V^l_k \). Also \( W^l_k \notin \pi_0 \). If \( W^l_k \) corresponds to \( V_k \) then \( V_k \) is true and clause \( C_i \) contains variable \( V_k \). If \( W^l_k \) corresponds to \( \overline{V}_k \), \( V_k \) is false and \( C_i \) contains the negation. So \( C_i \) is satisfied.

\( \Rightarrow \) Let \( V_i \in \{ \text{true, false} \} \) 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 \( C \). Let \( v(C_i) \in \{ V^1_j, V^2_j \} \) be the node corresponding to \( W \). Let \( p(i) = V^1_j \rightarrow \cdots \rightarrow V^o_j \) if \( V_i = \text{false} \), Otherwise let \( p(i) = V^1_j \rightarrow \cdots \rightarrow V^d_j \). We choose the following tuples which satisfy the conditions of Def. 3.2:

- \((Z_0, \{C_i, D_1 | 1 \leq i \leq n_C \}, Z_0 \leftarrow Z_0' \rightarrow V^1_i \rightarrow p(1) \rightarrow V^2_i \rightarrow p(2) \rightarrow V^3_i \rightarrow \cdots \rightarrow p(n_V) \rightarrow V^o_i \rightarrow X_0 \rightarrow Y)\),
- \((Z_1, \emptyset, Z_1 \rightarrow v(C_1) \rightarrow C_1 \rightarrow D_1 \rightarrow X_1 \rightarrow Y)\),
- \(\ldots\)
- \((Z_{n_C}, \emptyset, Z_{n_C} \rightarrow v(C_{n_C}) \rightarrow C_{n_C} \rightarrow D_{n_C} \rightarrow X_{n_C} \rightarrow Y)\).

(a) \( Y \) does not have any descendants and any \( \pi_i \) is an unblocked path connecting \( Z_i \) with \( X_{i} \rightarrow Y \).

(b) In \( \mathcal{G} \) all paths starting at \( Y \) begin with \( Y \leftrightarrow D_i \). In the first tuple the paths \( Y \leftrightarrow D_i \rightarrow Z_0 \) are blocked by \( D_i \) and the paths \( Y \leftrightarrow D_i \leftrightarrow C_i \) are blocked by \( C_i \). In all other tuples the paths \( Y \leftrightarrow D_i \rightarrow Z_0 \) are irrelevant and the paths \( Y \leftrightarrow D_i \leftrightarrow C_i \) are blocked by \( D_i \). No path \( \pi_i \) is blocked by \( W_i \).

(c) No path \( \pi_1, \ldots, \pi_{n_C} \) has a common node with \( \pi_0 \). Otherwise a node \( V^l_k \) 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.

\[\Box\]

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 \( O(nm) \) 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 \( O(nm) \) 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


