Learning Optimal Cyclic Causal Graphs from Interventional Data

Kari Rantanen KARI.RANTANEN@HELSINKI.FI
Antti Hyttinen ANTTI.HYTTINEN@HELSINKI.FI
Matti Järvisalo MATTI.JARVISALO@HELSINKI.FI
HIIT, Department of Computer Science, University of Helsinki, Finland

Abstract

We consider causal discovery in a very general setting involving non-linearities, cycles and several experimental datasets in which only a subset of variables are recorded. Recent approaches combining constraint-based causal discovery, weighted independence constraints and exact optimization have shown improved accuracy. However, they have mainly focused on the d-separation criterion, which is theoretically correct only under strong assumptions such as linearity or acyclicity. The more recently introduced sigma-separation criterion for statistical independence enables constraint-based causal discovery for non-linear relations over cyclic structures. In this work we make several contributions in this setting. (i) We generalize bcause, a recent exact branch-and-bound causal discovery approach, to this setting, integrating support for the sigma-separation criterion and several interventional datasets. (ii) We empirically analyze different schemes for weighting independence constraints in terms of accuracy and runtimes of bcause. (iii) We provide improvements to a previous declarative answer set programming (ASP) based approach for causal discovery employing the sigma-separation criterion, and empirically evaluate bcause and the refined ASP-approach.

Keywords: Graphical models; structure learning; causal discovery; exact search; optimization

1. Introduction

Discovering causal relations from sample data in general model spaces is a very challenging task. Approaches combining constraint-based causal discovery, weighted independence constraints and exact optimization developed within the last 5-10 years, based on harnessing both problem-specific and declarative techniques, have shown improved accuracy in the presence of latent variables and cycles over previous constraint-based approaches (Claassen and Heskes, 2012; Triantafillou and Tsamardinos, 2015; Hyttinen et al., 2014; Magliacane et al., 2016). However, most of these approaches are focused on the d-separation criterion, which is theoretically correct only under strong assumptions such as linearity or acyclicity.

In this work, we consider causal discovery, i.e., the task of learning optimal causal graphs, in a very general setting involving non-linearities, cycles (Richardson and Spirtes, 1999; Dash and Druzdzel, 2001; Hyttinen et al., 2012; Mooij and Heskes, 2013) and several experimental datasets in which (possibly) only a subset of variables are recorded (Tillman et al., 2009; Triantafillou et al., 2010; Tillman and Spirtes, 2011; Hyttinen et al., 2013; Triantafillou and Tsamardinos, 2015; Huang et al., 2020). The recently introduced σ-separation criterion for statistical independence enables constraint-based causal discovery for *both* non-linear relations and cyclic structures (Forré and Mooij, 2017). This allows for extending exact approaches to constraint-based causal discovery to non-linear settings. As a first concrete instantiation, Forré and Mooij (2018) recently extended an earlier *exact* declarative approach based on answer set programming (ASP) to causal discovery in the presence of cycles and experimental datasets to accommodate the σ-separation criterion[1].

In this work we take further the adaptation of both the ASP approach (Forré and Mooij, 2018) and a recent problem-specific branch-and-bound approach, *bcause* (Rantanen et al., 2020, 2018), to causal discovery under σ-separation over cyclic causal graphs from several interventional datasets. In addition to providing performance improvements to both approaches, we also aim to shed further light on their relative efficiency. Concretely, our contributions are the following. (i) We generalize *bcause* to this setting, in particular extending the approach with support for the σ-separation criterion and several interventional datasets. For the latter, we provide search-space pruning linear constraints between datasets which allow for obtaining tighter bounds via the linear relaxation based hitting-set approach implemented in *bcause*. (ii) We empirically analyze different schemes for weighting independence constraints in terms of accuracy and runtimes of *bcause*, revealing that the choice of the weighting scheme appears to have a noticeable impact on both accuracy and efficiency of causal discovery, with a trade-off between accuracy and efficiency. (iii) We refine the ASP-based approach for causal discovery under σ-separation, and show that the modifications improve on the runtime efficiency of the approach. (iv) Finally, we empirically evaluate the runtime performance of the *bcause* and ASP approaches, showing that the relative efficiency of the approaches depends even significantly on the data considered.

2. Constraint-Based Causal Discovery

Directed Mixed Graphs We represent causal structure by a directed mixed graph $G = (V, E)$ over the set of nodes V, where the edge relation $E = E_{\rightarrow} \cup E_{\leftrightarrow}$ is composed of directed edges $E_{\rightarrow} \subseteq V \times V$ and (symmetric) bi-directed edges $E_{\leftrightarrow} \subseteq \{\{X, Y\} : X, Y \in V\}$. A bi-directed edge $X \leftrightarrow Z$ represents a *latent confounder*, i.e., a structure $X \leftarrow L \rightarrow Z$, where L is an unmeasured common cause of two observed variables X and Z. A walk in a DMG G is any sequence of adjacent edges, with repetitions allowed. A triple of adjacent nodes (V_{i-1}, V_i, V_{i+1}) on a walk of the form

$$V_{i-1} \rightarrow V_i \leftarrow V_{i+1}, \quad \text{or} \quad V_{i-1} \leftrightarrow V_i \leftarrow V_{i+1} \quad \text{or} \quad V_{i-1} \rightarrow V_i \leftrightarrow V_{i+1} \quad \text{or} \quad V_{i-1} \leftrightarrow V_i \leftrightarrow V_{i+1}$$

is called a *collider* (both edges pointing into V_i). Other triples of adjacent nodes on a walk are called *non-colliders* (at least one edge out of V_i). The class of directed mixed graphs is denoted by $\mathcal{G}$.

Separation Criteria In order to perform constraint-based causal discovery we need a graphical criterion for independence. If the underlying causal model is a linear structural equation model and the data is observed at a unique equilibrium, then the following d-separation criterion is sufficient for independence (Spirtes, 1995) (see Studený (1998) for equivalence to the version of Pearl (2000)).

Definition 1 (d-separation) *Let G be a DMG over nodes V. Given a conditioning set $C \subseteq V$, a walk π between $V_1, V_n \notin C$ is* d-connecting*, if every triple of adjacent nodes in π is:*

(a) a collider satisfying $V_i \in C$; or *(b) a non-collider satisfying $V_i \notin C$.*

Nodes are d-connected given C if and only if there is a d-connecting walk given C between them.

[1] Recently, Mooij and Claassen (2020) develop *in-exact* methods employing σ-separation.

However, when we have non-linear relations and cycles, d-separated variables may be dependent (Spirtes, 1995; Neal, 2000). To this end, Mooij et al. proposed the notion of σ-separation which is sound for non-linear relations under global compatibility of the local causal mechanisms (see (Forré and Mooij, 2018)). Given a DMG G, we use $Sc^G(V_i)$ to denote the set of nodes that are strongly connected to V_i, that is, each node in $Sc^G(V_i)$ is both a descendant and an ancestor of V_i.

Definition 2 (σ-connection, Forré and Mooij (2019) Definition 4.2) *Let G be a DMG over a set of nodes V. Given a conditioning set $C \subseteq V$, a walk π between $V_1, V_n \notin C$ is* σ-connecting*, if every triple of adjacent nodes (V_{i-1}, V_i, V_{i+1}) in π is:*

(a) a collider satisfying $V_i \in C$; or
(b) $V_{i-1} \leftarrow V_i \leftarrow V_{i+1}$ or $V_{i-1} \leftarrow V_i \leftrightarrow V_{i+1}$, satisfying $V_i \notin C$ or $V_i \in C \cap Sc^G(V_{i-1})$; or
(c) $V_{i-1} \rightarrow V_i \rightarrow V_{i+1}$ or $V_{i-1} \leftrightarrow V_i \rightarrow V_{i+1}$, satisfying $V_i \notin C$ or $V_i \in C \cap Sc^G(V_{i+1})$; or
(d) $V_{i-1} \leftarrow V_i \rightarrow V_{i+1}$ satisfying $V_i \notin C$ or $V_i \in C \cap Sc^G(V_{i-1}) \cap Sc^G(V_{i+1})$.

Nodes are σ-connected given C if and only if there is a σ-connecting walk given C between them.

For example, in Figure 1 the nodes X, Y are σ-connected given Q, W as e.g. the walk $X \rightarrow Q \rightarrow Y$ satisfies condition (c) of Definition 2. Nodes X, Y are d-separated given Q, W. By assuming σ-faithfulness (faithfulness) statistical independence and σ-separation (d-separation) become equivalent (Forré and Mooij, 2018).

Problem Definition In constraint-based causal discovery the aim is to find an equivalence class of graphs whose separation and connection properties respectively match the statistical independence and dependence relations in the data. The (in)dependence constraints K are obtained from statistical independence tests on the data. Since the tests produce some errors on finite sample data, constraint-based causal discovery corresponds to the following optimization problem (Hyttinen et al., 2014).

INPUT: A set K of conditional (in)dependence constraints over given set of variables V, and a non-negative weight $w(k)$ for each $k \in K$.

TASK: Find a causal graph $G^* = (V, E^*)$ such that

$$G^* \in \operatorname{argmin}_{G \in \mathcal{G}} \sum_{k \in K \,:\, G \not\models k} w(k). \tag{1}$$

In words, our goal is to find a single graph G^* that minimizes the sum of the weights of the (in)dependence constraints *not* implied ($\not\models$) by G^*. The weight function $w(\cdot)$ describes the reliability of each constraint (obtained by independently run tests): conflicts among the constraints are well-resolved when the sum of the weights of the constraints not satisfied is minimized. Since the score function trivially satisfies score equivalence (Heckerman et al., 1995), an optimal causal graph G^* is a representative of the (Markov) equivalence class closest to the input constraints.

There are a number of different ways to obtain reliability weights for the independence constraints (see Section 5). Apart from a constraint satisfaction perspective, the objective function is equivalent to maximizing the posterior probability $P(G|D)$ under the simplifying modelling assumptions: a) constraints are distributed independently given D, and b) (in)dependence constraints exhaust all information on the causal graph in D (Jabbari et al., 2017; Hyttinen et al., 2014).

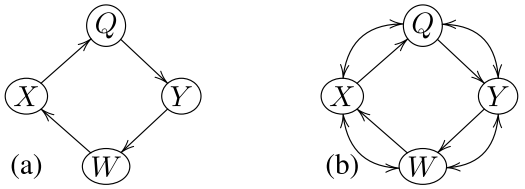

Figure 1: (a) A directed mixed graph (DMG). (b) The σ-extension of the DMG in (a).

3. Extending the *bcause* Approach to Causal Discovery

The recent *bcause* approach (Rantanen et al., 2020), based on branch and bound (van Beek and Hoffmann, 2015; Suzuki, 1996) with problem-specific techniques for speeding up search, offers a competitive way of exactly solving the causal discovery problem under d-separation. In the following, we extend the approach to support σ-separation and interventional datasets.

The branch-and-bound search in *bcause* is performed over the edge relation of DMGs with cycles. Each node in the search tree corresponds to some partial solution (a DMG with some edges possibly undecided). Starting from an empty partial solution with every edge undecided, edges are decided recursively in depth-first manner as either present or absent. That is, two new branches are opened at each point in the tree (excluding leaf nodes), which correspond to new edge decisions. The leaves of the search tree have no undecided edges remaining and thus they represent all the possible solutions (DMGs) for the given problem instance. As the search keeps track of the lowest-weight solution G^* encountered so far in the search, G^* will be optimal solution once the search has finished. Instead of having to construct the entire search tree, *bcause* employs strong core-based lower bounding, as discussed later on, for identifying provably suboptimal partial solutions and pruning the corresponding branches out from the search tree.

3.1 Implementing the σ-separation Criterion

We continue by explaining how σ-separation can be integrated into *bcause*. We will use the following equivalent definition of σ-separation as it corresponds more closely to that of d-separation. Compared to Definition 2, all cases where $V_i \notin C$ are now in item (b), and cases where $V_i \in C$ are gathered in item (c).

Definition 3 (σ-connection) *Let G be a DMG over a set of nodes V. Given a conditioning set $C \subseteq V$, a walk π between $V_1, V_n \notin C$ is* σ-connecting, *if every triple of adjacent nodes (V_{i-1}, V_i, V_{i+1}) in π is:*

(a) a collider satisfying $V_i \in C$; or (b) a non-collider satisfying $V_i \notin C$; or
(c) a non-collider satisfying $V_i \in C$, and if $V_i \to V_{i+1}$ in the triple, then $V_i \in Sc^G(V_{i+1})$,
and if $V_i \to V_{i-1}$ in the triple, then $V_i \in Sc^G(V_{i-1})$.

Nodes are σ-connected given C if and only if there is σ-connecting walk given C between them.

The first two conditions of Definition 3, (a) and (b), together form the definition for d-connection (Definition 1), and only the condition (c) is σ-connection-specific. This motivates us to consider the following σ-extension, which allows to check σ-separation through d-separation. This is similar transformation as acyclification (Forré and Mooij, 2017; Forré and Mooij, 2018).

Definition 4 (σ-extension) *Let $G = (V, E)$ be a DMG. The σ-extension of G, G' contains all the edges in G and for any edge $V_i \to V_j \in G$ such that $V_i \in Sc^G(V_j)$, G' includes also $V_i \leftrightarrow V_j$.*

Note that the the bi-directed edges added at the last step of the σ-extension definition, corresponds directly to condition (c) of Definition 3. The connection between d-separation and σ-separation can be now confirmed as the following theorem.

Theorem 5 *Nodes X,Y are σ-separated (connected) given C in G if and only if they are d-separated (connected) given C in the σ-extension of G.*

Proof Suppose walk π is σ-connecting given C in G by Definition 3. Based on this let us produce a d-connecting path π' in the σ-extension of G. We need to replace the parts where there is a triple (V_{i-1}, V_i, V_{i+1}) satisfying the requirements of (c). This means that V_i is not a collider and $V_i \in C$. If $V_i \to V_{i+1}$ we have $V_i \in Sc^G(V_{i+1})$, in the σ-extension there would be $V_i \leftrightarrow V_{i+1}$. If $V_i \to V_{i-1}$ we have $V_i \in Sc^G(V_{i-1})$, in the σ-extension there would be $V_i \leftrightarrow V_{i-1}$. Replacing the directed edge(s) with the added bi-directed edge(s) ensures d-connection since V_i is conditioned on. This change is local: it does not affect d-connectivity through other nodes.

Then suppose there is a walk π' that is d-connecting in the σ-extension and let us produce a walk π that is σ-connecting. The walk π' may go through edges $V_i \leftrightarrow V_{i+1}$ that are in the extension but not present in G. Edge $V_i \leftrightarrow V_{i+1}$ can only be added to the extension if there $V_i \to V_{i+1}$ or $V_{i+1} \to V_i$. Without loss of generality assume G has $V_i \to V_{i+1}$ and thus $V_i \in Sc^G(V_{i+1})$ by Definition 4. Then, $V_i \leftrightarrow V_{i+1}$ can be replaced by $V_i \to V_{i+1}$ on the path. If V_i is not conditioned on, walk comes to V_i with a tail and σ-connection is ensured by Definition 3(b). If V_i is conditioned on, the requirements of Definition 3(c) are fulfilled and the walk is σ-connecting. ■

For example, X, Y can be confirmed to be σ-connected given Q, W in Figure 1 (a) since in the extension shown in Figure 1 (b), $X \leftrightarrow Q \leftrightarrow Y$ is a d-connecting walk given Q, W.

When considering d-separation, *bcause* checks whether a d-connecting walk exists in a graph between given two nodes X and Y and a conditioning set C by starting from X and moving from node to node through the edges of the graph until we reach Y, excluding all non-active walks along the way. Concretely, let Z and Q be nodes in the graph. If $Q \in C$, we are not allowed to move through an edge $Z \leftarrow Q$. Additionally, if $Z \notin C$ and we had moved into Z through an inward edge ($\to Z$ or $\leftrightarrow Z$), we are not allowed to move through an edge $Z \leftarrow Q$ or $Z \leftrightarrow Q$ unless there is a directed path $Z \to V_1 \to \ldots \to V_n$ where $V_1, \ldots, V_{n-1} \notin C$ and $V_n \in C$. When using σ-separation, the previous rules can be extended with the following exception: if $Q \in C$, we can move through an edge $Z \leftarrow Q$ if there exists a directed path from Z to Q.

3.2 σ-separation and Core-based Lower Bounds

As detailed in (Rantanen et al., 2020), *bcause* uses so-called core-based lower bounding during search via linear relaxations. We now establish the correctness of this approach for σ-separation.

An *unsatisfiable core* is a set of (in)dependence constraints that cannot be simultaneously satisfied by any graph in $\mathcal{G}$. One example is $\{X \not\perp\!\!\!\perp Y, X \not\perp\!\!\!\perp Z, Y \perp\!\!\!\perp Z | X, Y \perp\!\!\!\perp Z\}$: the dependencies of X on Y and Z imply a dependence between Y and Z either marginally or conditional on X. To find cores for the input dataset in the beginning of the search, *bcause* uses the seven core patterns from (Hyttinen et al., 2017). Using these, lower bounds are obtained during search by formulating a minimum-cost hitting set problem where the unsatisfiability cores represent the sets and the (in)dependence constraints represent the elements. The objective is then to find a minimum-cost subset of constraints that contains something from each core. To obtain the bounds in practice, the linear relaxation of a standard integer programming formulation of these hitting set problems using a linear programming (LP) solver.

The following theorem validates the use of cores (Hyttinen et al., 2017) obtained with d-separation inside each experimental datasets also when using σ-sepration.

Proposition 6 *Unsatisfiable cores of Hyttinen et al. (2017) are correct when using the σ-separation interpretation.*

Proof Since for a given core of constraints there is no DMG with d-separation interpretation, there cannot be a DMG with σ-separation interpretation (Theorem 5). ■

3.3 Extension to Interventional Data

In addition to integrating σ-separation, we extend *bcause* to support interventional data. Furthermore, extending the cores of Hyttinen et al. (2017), we describe further LP constraints which capture relationships *between* different experimental datasets.

To integrate support for interventional data into *bcause*, we modify the d-connection/σ-connection checking within *bcause*. In particular, as explained at the end of Section 3.1, *bcause* checks for the existence of a relevant d-connecting (or σ-connecting) walk when determining whether a constraint is satisfied in a graph. This allows for taking interventions into account by assuming the absence of incoming edges to nodes that are in the intervention set when searching for the walk.

The cores of Hyttinen et al. (2017) applied originally in *bcause* are semantically restricted to a single data set. However, when there are several experimental datasets one can find additional cores *across* datasets. One example of such a core is $\{X \perp\!\!\!\perp Y, X \not\perp\!\!\!\perp Y||Z\}$ where the notation $||$ means that the second constraint is obtained when intervening on Z. The core is due to the fact that intervening on Z removes edges, and thus separation cannot turn to a connection. In general, for any variables X, Y and a conditional set C, we have the following logical implications on constraints across different experimental datasets:

$$\begin{aligned} X \perp\!\!\!\perp Y|C||J &\Rightarrow X \perp\!\!\!\perp Y|C||J', && (2)\\ X \not\perp\!\!\!\perp Y|C||J' &\Rightarrow X \not\perp\!\!\!\perp Y|C||J, && (3)\end{aligned}$$

where the intervention sets satisfy $J' \supset J$. For example, consider the graph (a) in Figure 2. Clearly, a constraint $X \not\perp\!\!\!\perp Y|Z,W||\emptyset$ is violated in the graph. By the rule stated above, a constraint $X \not\perp\!\!\!\perp Y|Z,W||Q$ has to be violated as well, since removing the incoming edge to Q would not increase the connectivity between X and Y. In the LP used for core-based lower bounding the above rules can be expressed as the LP constraints

$$\begin{aligned} (X \perp\!\!\!\perp Y|C||J) &\leq (X \perp\!\!\!\perp Y|C||J'),\\ (X \perp\!\!\!\perp Y|C||J) &\leq 1-(X \not\perp\!\!\!\perp Y|C||J'),\\ 1-(X \not\perp\!\!\!\perp Y|C||J) &\leq (X \perp\!\!\!\perp Y|C||J'),\\ 1-(X \not\perp\!\!\!\perp Y|C||J) &\leq 1-(X \not\perp\!\!\!\perp Y|C||J'),\end{aligned}$$

where each $(\cdot)$ denotes a variable in the LP such that it is 1 if the constraint is satisfied and 0 if it is not. (Due to the LP relaxation the variables may take values from $[0, 1]$.)

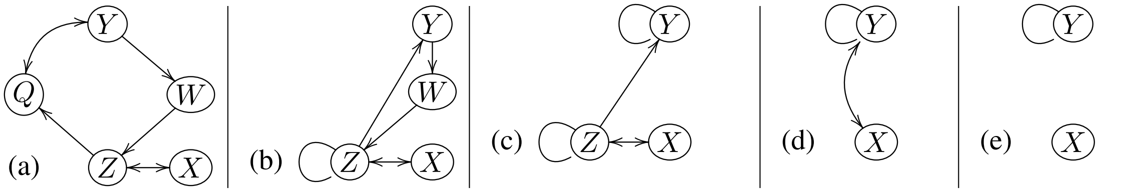

Figure 2: (a) Original DMG. Rest are σ-connection graphs: (b) when Q conditioned on (c) when also W is conditioned on. Graph (d) shows the result of the original code when also Z is conditioned on. Graph (e) is the corrected version when Z is conditioned on.

4. Improving the ASP-based Approach of Forré and Mooij (2018)

We turn to the ASP-based discovery approach of Forré and Mooij (2018) for non-linear cyclic models that is a modification the original ASP-based approach of Hyttinen et al. (2014) using the d-separation criterion. The approach declaratively encodes the σ-separation criterion in the ASP language, and applies the Clingo ASP solver for exact optimization. We suggest changes to the ASP encoding to improve the overall efficiency of the approach, and fix an issue in their encoding which in cases results in non-optimal/wrong solutions. We will only detail our technical changes to the encoding of Forré and Mooij (2018); please refer to their paper for full details on the encoding.

The encoding of Forré and Mooij (2018) builds the connections of (in)dependence constraints and the causal graph structure using operations of marginalization, conditioning and intervention of a single node at a time, following the encoding for d-separation by Hyttinen et al. (2014). The idea is that from the causal graph one can apply these operations node at a time, maintaining connection properties on the way, and arrive at a graph over just two nodes mentioned in the relations: if there is an arc, then the nodes are dependent, otherwise they are independent. Such an approach shares information among the constraints and for d-separation achieves considerable speedups compared to a path-specific encoding of Hyttinen et al. (2013) (Hyttinen et al., 2014). Forré and Mooij (2018) encodes as ASP rules the formation of the tree (see Hyttinen et al. (2014) Figure 3) of operations at grounding phase. However, we found that their tree is overly complex and the following simplified version achieves marked running time improvements.

```
node(0..nrnodes-1). set(0..2**nrnodes-1).
ismember(M,Z) :- set(M),node(Z),M & (2**Z) != 0.

%mark the cjm combinations present at input
cjm(C,J,M) :- indep(X,Y,C,J,M,W).
cjm(C,J,M) :- dep(X,Y,C,J,M,W).

%first take out marginalization
% condition 2**Z > M-2**Z means Z is the largest variable in M
marginalize(C,J,M-2**Z,Z,M) :- cjm(C,J,M),ismember(M,Z),2**Z > M-2**Z,node(Z).
cjm(C,J,Msub) :- marginalize(C,J,Msub,Z,M).

%then take out conditioning
condition(C-2**Z,Z,C,J,0) :- cjm(C,J,0),ismember(C,Z), 2**Z > C-2**Z,node(Z).
cjm(Csub,J,0) :- condition(Csub,Z,C,J,0).

%finally take out intervention
```

```
intervene(0,J-2**Z,Z,J,0) :- cjm(0,J,0),ismember(J,Z),2**Z > J-2**Z,node(Z).
cjm(0,Jsub,0) :- intervene(0,Jsub,Z,J,0).
```

Furthermore, through extensive experimentation we found inconsistencies the implementation of Forré and Mooij (2018)[2]. With the use of Theorem 5, we were able to narrow down the problematic cases. Consider the left most graph[3] in Figure 2 and whether X is σ-separated from Y given Q, W, Z. We can confirm that the corresponding σ-separation holds using Definition 2 (or since the graph is acyclic, using the d-separation in Definition 1). The encoding of Forré and Mooij (2018) checks the state of this constraints by applying conditioning operations to Q, W, and Z respectively, as shown in Figure 2 (b-d). When conditioning on Z in Figure 2 (c) and arc $X \leftrightarrow Y$ is drawn (Figure 2 (d)) due the rule marked with `%% X<->Z-->Y (anc of Z) => X<->Y (sigma)` (lines 125–133 of the ASP code) suggesting that X and Y are σ-connected given Z, Q, W. This is because the rule only requires `ancestor(Y,Z,J)`, which is satisfied as Y is an ancestor of Z in the original graph. Contrary to this, Definition 2 (or Definition 2.19 in Forré and Mooij (2018)) requires that Y and Z should be in the same strongly connected component (which means Y and Z should be ancestors of each other). Here, in Figure 2 (a), Z is not an ancestor Y (even though there is a tail-head path in Figure 2 (c)). Thus adding `ancestor(Z,Y,J)` to the body (right-hand side) of the rule and several other rules similarly prevents this behaviour. This corrected encoding produces Figure 2 (e) when conditioning on Z in Figure 2 (c), and correctly implies that $X \not\perp\!\!\!\perp Y | Z, Q, W$. We have tested the corrected encoding against *bcause+*, i.e., our extension of *bcause* to σ-separation and interventional data, and have found no deviations in the costs of optimal solutions reported.

5. EXPERIMENTS

We provide empirical results from three perspectives: (i) the impact of weighting schemes on accuracy and runtime efficiency; (ii) the impact of the novel constraints (as described in Section 3.3) on the runtimes of *bcause+*; and (iii) the relative runtime efficiency of *bcause+*, the ASP approach of (Forré and Mooij, 2018), and our modification ASP+ of the approach of Forré and Mooij (2018).

Impact of Weighting Schemes We consider three weighting schemes: a Bayesian model selection based scheme, where weights are the difference of local scores $w = |s(Y, \{X\} \cup C) - s(Y, C)|$, where $s(\cdot, \cdot)$ denotes the local score familiar from Bayesian network structure learning ('BDeu' with different ESS, or BGe') (Hyttinen et al., 2014); frequentist independence test based weights, where $w = |\log(p) - \log(p_{\text{threshold}})|$ and p denotes the p-value obtained in a particular test and $p_{\text{threshold}}$ is the threshold used ('Freq') (Magliacane et al., 2016); and posterior probabilities of d-separation by averaging over DAGs for the variables involved in the test ('Pdsep') (Claassen and Heskes, 2012).[4]

We used 100 causal graphs over 6 nodes. Edges were sampled randomly with average node-degree 3. We generated data from three partially overlapping experimental datasets: one passive observational and two where one variable was intervened on and one was unobserved, with 300 samples each. For simplicity, we assumed here acyclicity and joint causal sufficiency. Joint causal sufficiency allows for latent confounding inside each data set but prohibits it wrt the full set of nodes V, thus translating to the absence of bidirected edges. As suggested by Magliacane et al. (2016), we

[2] We refer here to version 1.1 of the implementation. The corrections pointed out here have subsequently been implemented in version 1.2.

[3] Figure 5 in the supplementary material of Forré and Mooij (2018) shows a similar example.

[4] We note that further proposed approaches are variations of these (Cooper, 1997; Margaritis and Bromberg, 2009; Triantafillou and Tsamardinos, 2015; Jabbari et al., 2017).

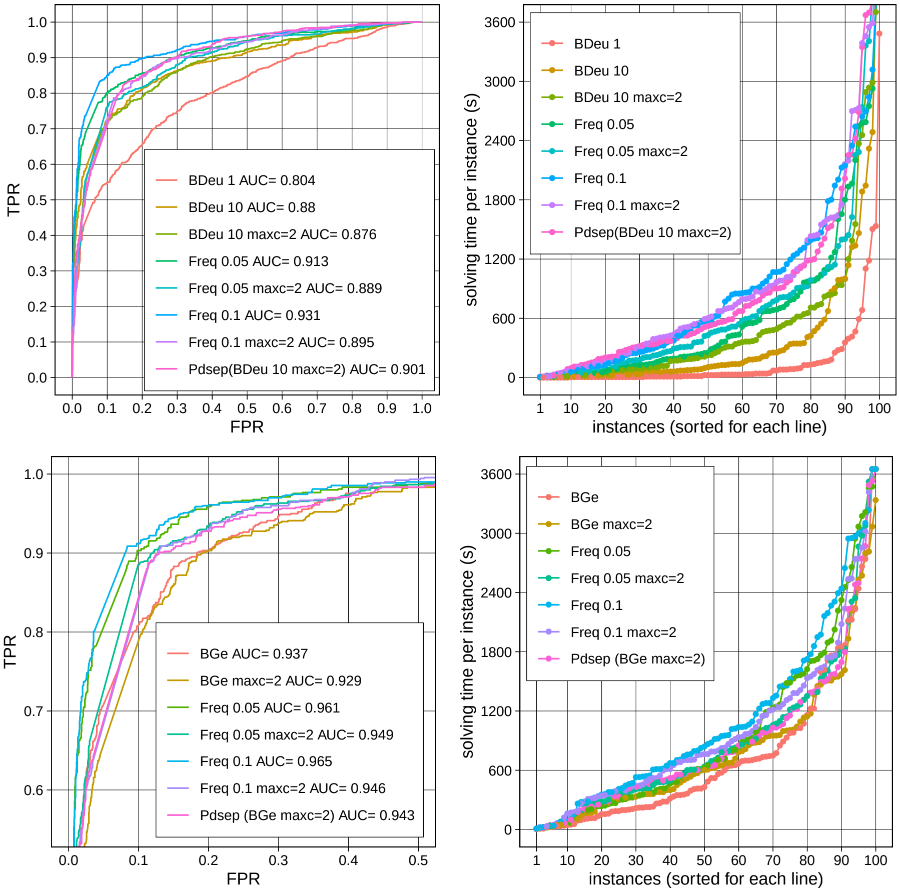

Figure 3: ROC-curves and total running time on discrete and continuous data.

scored the presence of each edge in the graphs by the difference of score of the optimal graph with the edge present and the optimal graph with the edge absent. One benefit of this is that edge scores are deterministically determined by the data; there is no bias due to solvers favouring particular type of graphs over others.

Figure 3 (top left) shows the ROC curve for the different weights. Note that that the straight parts on any curve are due to several edges receiving zero score, meaning that the data does not provide any support in either existence or absence of them. The frequentist based weights exhibit best performance when the p-value threshold is at 0.1, suggesting that care should be taken when selecting this threshold for the sample sizes used in each data set. Bayesian model selection based scores exhibit weaker performance here. This is partially due to high complexity penalities overweighing moderate dependencies in tests with larger conditioning sets. Figure 3 (top right) gives the total running times of *bcause* for obtaining all edge scores for each of the 100 instances. Limiting the conditioning set size for the test does not have a great effect; including more may actually make solving faster (due to better bounds). BDEU 1 is the fastest to solve, but also results in worst accuracy. Figure 3 (bottom row) shows similar results for continuous linear Gaussian data. Overall, for higher accuracy longer solving times are needed. Frequentist tests offer best accuracy, although this hinges on choosing the right p-value threshold for the particular sample size.

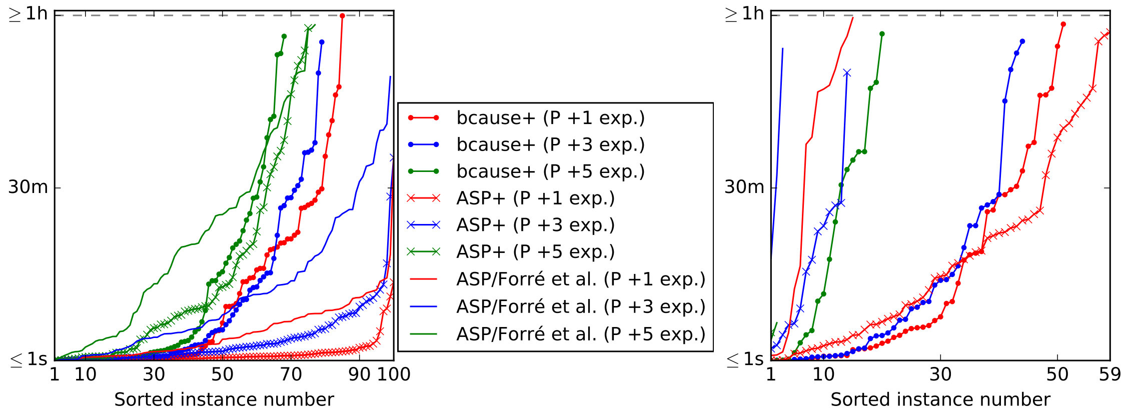

Figure 4: *bcause*+, ASP (by Forré and Mooij (2018)) and ASP+ (our modification of ASP) on instances with 1 (“P +1 exp”), 3 (“P +3 exp”) and 5 (“P +5 exp”) experimental datasets. *Left*: 6-node graphs. *Right*: 7-node graphs under joint causal sufficiency assumption.

Runtime Efficiency We evaluate the runtime performance of *bcause*+ and *ASP*+ against the ASP approach of Forré and Mooij (2018) on simulated data from non-linear cyclic causal models generated following Forré and Mooij (2018). With 300 samples per dataset, the data was simulated with 6-7 observed nodes and 2 latent nodes, the nonlinear relationships were neural network based. The average node-degrees were 2.2 (6 observed nodes) and 2.3 (7 observed nodes). Constraint weights were obtained from an independence test based on rank transformations with threshold 0.01 and with conditioning set size limited to three. 6-node instances with 1, 3 and 5 experimental datasets resulted in 450, 900 and 1350 constraints, respectively, and 7-node instances with 1, 3 and 5 experimental data sets in 1092, 2184 and 3276 constraints, respectively.

Figure 4 shows the number of instances solved to optimum (x-axis) under different per-instance time limits (y-axis). Our enhanced ASP+ is consistently the fastest of the three approaches on 6-variable instances (left). Here both ASP encodings performed better than *bcause*+, although the performance gap narrows as the number of experimental datasets is increased. On 7-variable instances under joint causal sufficiency (right), the enhanced ASP+ outperforms the rest on instances with one experimental dataset. However, *bcause*+ significantly outperforms both of the ASP approaches as the number of experimental datasets is increased; we expect this to be due to the fact that the number of constraints increases significantly as the number of datasets is increased, which results in larger declarative encodings hindering the ASP solver. Furthermore, our enhanced ASP+ consistently outperforms the original ASP approach by Forré and Mooij (2018). Finally, as seen in Figure 5, we observe that the additional cores between datasets (recall Section 3.3) give a non-negligible boost to the runtime performance of *bcause*+ for each experimental setting.

6. CONCLUSION

We generalized and improved recent exact approaches to causal discovery in a very general setting, involving non-linearities, cycles and several experimental datasets in which only a subset of variables are recorded. In particular, we generalized a recent branch-and-bound approach to causal discovery by integrating support for σ-separation and interventional data, and provided improve-

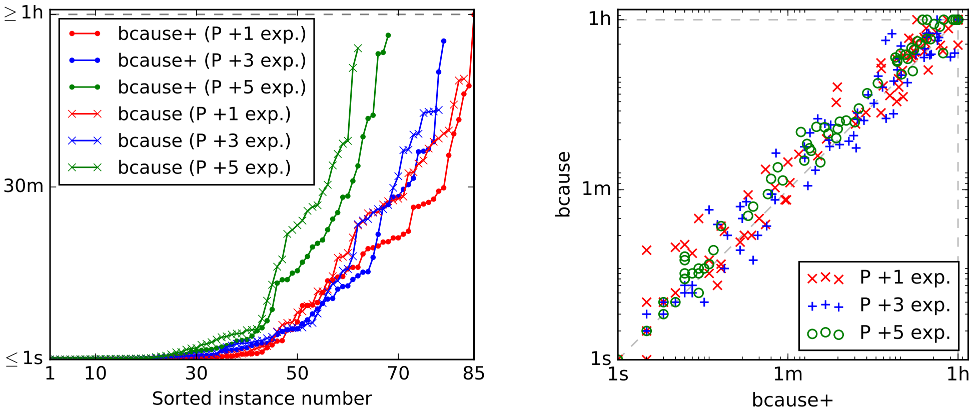

Figure 5: Comparison of *bcause* (*bcause*+ without dataset linking LP constraints) and *bcause*+.

ments to an ASP-based declarative approach to causal discovery in this setting. Empirically, we observed a tradeoff in accuracy vs runtimes of exact causal discovery brought on by the choice of weighting schemes for independence constraints, and further showed that, depending on the data at hand, both the generalized branch-and-bound approach and the improved ASP approach can even significantly outperform a recent ASP approach to causal discovery under σ-separation.

Acknowledgments

This work was supported by Academy of Finland via grants 295673, 316771, 322869, 328718.

References

T. Claassen and T. Heskes. A Bayesian approach to constraint based causal inference. In *UAI*, pages 207–216. AUAI Press, 2012.

G. F. Cooper. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Min. Knowl. Discov.*, 1(2):203–224, 1997.

D. Dash and M. Druzdzel. Caveats for causal reasoning with equilibrium models. In *ECSQARU*, volume 2143 of *LNCS*, pages 192–203. Springer, 2001.

P. Forré and J. M. Mooij. Markov properties for graphical models with cycles and latent variables. *arXiv.org preprint*, arXiv:1710.08775 [math.ST], 2017.

P. Forré and J. M. Mooij. Constraint-based causal discovery for non-linear structural causal models with cycles and latent confounders. In *UAI*, pages 269–278. AUAI Press, 2018.

P. Forré and J. M. Mooij. Causal calculus in the presence of cycles, latent confounders and selection bias. In *UAI*, pages 15–24. AUAI Press, 2019.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

B. Huang, K. Zhang, M. Gong, and C. Glymour. Causal discovery from multiple data sets with non-identical variable sets. In *AAAI*, 2020.

A. Hyttinen, F. Eberhardt, and P. O. Hoyer. Causal discovery of linear cyclic models from multiple experimental data sets with overlapping variables. In *UAI*, pages 387–396, 2012.

A. Hyttinen, P. O. Hoyer, F. Eberhardt, and M. Järvisalo. Discovering cyclic causal models with latent variables: A general SAT-based procedure. In *UAI*. AUAI Press, 2013.

A. Hyttinen, F. Eberhardt, and M. Järvisalo. Constraint-based causal discovery: Conflict resolution with answer set programming. In *UAI*, pages 340–349. AUAI Press, 2014.

A. Hyttinen, P. Saikko, and M. Järvisalo. A core-guided approach to learning optimal causal graphs. In *IJCAI*, pages 645–651. ijcai.org, 2017.

F. Jabbari, J. Ramsey, P. Spirtes, and G. F. Cooper. Discovery of causal models that contain latent variables through Bayesian scoring of independence constraints. In *ECML-PKDD*, volume 10535 of *LNCS*, pages 142–157. Springer, 2017.

S. Magliacane, T. Claassen, and J. M. Mooij. Ancestral causal inference. In *NIPS*, pages 4466–4474. Curran Associates, 2016.

D. Margaritis and F. Bromberg. Efficient Markov network discovery using particle filters. *Computational Intelligence*, 25(4):367–394, 2009.

J. M. Mooij and T. Claassen. Constraint-based causal discovery using partial ancestral graphs in the presence of cycles. In *UAI*, 2020.

J. M. Mooij and T. Heskes. Cyclic causal discovery from continuous equilibrium data. In *UAI*, pages 431–439. AUAI Press, 2013.

R. Neal. On deducing conditional independence from d-separation in causal graphs with feedback. *Journal of Artificial Intelligence Research*, 12:87–91, 2000.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

K. Rantanen, A. Hyttinen, and M. Järvisalo. Learning optimal causal graphs with exact search. In *PGM*, volume 72 of *Proceedings of Machine Learning Research*, pages 344–355. PMLR, 2018.

K. Rantanen, A. Hyttinen, and M. Järvisalo. Discovering causal graphs with cycles and latent confounders: An exact branch-and-bound approach. *Int. J. Approx. Reason.*, 117:29–49, 2020.

T. Richardson and P. Spirtes. Automated discovery of linear feedback models. In C. Glymour and G. F. Cooper, editors, *Computation, Causation & Discovery*, pages 253–302. MIT Press, 1999.

P. Spirtes. Directed cyclic graphical representations of feedback models. In *UAI*, pages 491–498. Morgan Kaufmann, 1995.

M. Studený. Bayesian networks from the point of view of chain graphs. In *Proc. UAI*, pages 496–503. Morgan Kaufmann, 1998.

J. Suzuki. Learning Bayesian belief networks based on the minimum description length principle: An efficient algorithm using the B & B technique. In *ICML*, pages 462–470, 1996.

R. E. Tillman and P. Spirtes. Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In *AISTATS*, volume 15 of *JMLR W&CP*, pages 3–15, 2011.

R. E. Tillman, D. Danks, and C. Glymour. Integrating locally learned causal structures with overlapping variables. In *NIPS*, pages 1665–1672, 2009.

S. Triantafillou and I. Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *J. Mach. Learn. Res.*, 16:2147–2205, 2015.

S. Triantafillou, I. Tsamardinos, and I. G. Tollis. Learning causal structure from overlapping variable sets. In *AISTATS*, pages 860–867. JMLR, 2010.

P. van Beek and H. Hoffmann. Machine learning of Bayesian networks using constraint programming. In *CP*, volume 9255 of *LNCS*, pages 429–445, 2015.