Journal of Machine Learning Research 24 (2023) 1-35 Submitted 9/22; Revised 5/23; Published 8/23

Scalable Computation of Causal Bounds

Madhumitha Shridharan MS6143@COLUMBIA.EDU

Garud Iyengar GI10@COLUMBIA.EDU

Department of Industrial Engineering and Operations Research
Columbia University
New York City, NY 10027, USA

Editor: Jin Tian

Abstract

We consider the problem of computing bounds for causal queries on causal graphs with unobserved confounders and discrete valued observed variables, where identifiability does not hold. Existing non-parametric approaches for computing such bounds use linear programming (LP) formulations that quickly become intractable for existing solvers because the size of the LP grows exponentially in the number of edges in the causal graph. We show that this LP can be significantly pruned, allowing us to compute bounds for significantly larger causal inference problems compared to existing techniques. This pruning procedure allows us to compute bounds in closed form for a special class of problems, including a well-studied family of problems where multiple confounded treatments influence an outcome. We extend our pruning methodology to fractional LPs which compute bounds for causal queries which incorporate additional observations about the unit. We show that our methods provide significant runtime improvement compared to benchmarks in experiments and extend our results to the finite data setting. For causal inference without additional observations, we propose an efficient greedy heuristic that produces high quality bounds, and scales to problems that are several orders of magnitude larger than those for which the pruned LP can be solved.

Keywords: Causal Bounds, Partial Identification of Causal Effects, Causal Bounds with Observations, Multi-Cause Setting with Unobserved Confounders, Linear Programming

1. Introduction

We are interested in answering the following counterfactual query about a large-scale system of discrete variables: What will be the value of some outcome variables V_O if we intervene on variables V_I, given the values of variables V_A are known? Several meaningful questions in data-rich environments can be formulated this way. As an example, consider the causal graph in Figure 1 that describes the health outcome of a patient Y as a function of a sequence of treatments administered T_i by a physician, $i = 1, \ldots, 5$. The treatments chosen are functions of some patient characteristics C_i, $i = 1, 2$, e.g. sex and age. The variable U_B refers to unknown variables, also known as confounders, that might impact both the choice of treatments and the health outcome e.g. patient lifestyle, physician biases, etc. What will be the expected health outcome Y of the patient if she is administered treatments $T_i, i = 1, \ldots, 5$, given her sex and age are known? (Ranganath and Perotte, 2019; Janzing and Schölkopf, 2018; D'Amour, 2019; Tran and Blei, 2017)

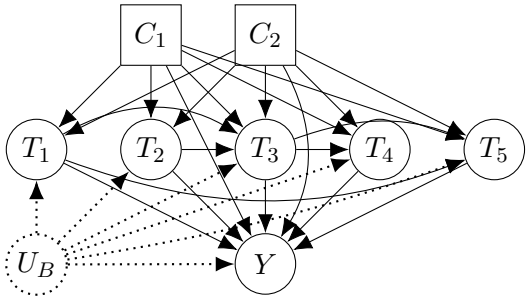


Figure 1: C_1 and C_2 denote the sex and age of the patient. These influence the treatments, $T_i, i = 1 \dots 5$, which the patient is prescribed. Note that the prescription of one treatment can influence the prescription of another (e.g. T_3 is prescribed to manage the side effects of T_1). Together, these variables influence Y, the health outcome of the patient. Unobserved confounder U_B denotes unobserved variables, e.g. patient lifestyle, physician biases, that impact the choice of treatments, and the health outcome.

Traditional approaches to estimate causal effects of interventions involve randomized control trials (RCTs) in order to remove the impact of confounders. However, running experiments to identify personalized interventions for sub-populations of units is often expensive and practically infeasible. Therefore, there is a push to develop techniques that can use observational data that is far more readily available.

The challenge in observational studies is to account for unobserved confounders which can create spurious correlations and adversely impact data-driven decision-making (Imbens and Rubin, 2015; Pearl, 2009). For example, the unknown confounder U_B in Figure 1 influences both the prescribed treatments and the outcome: a patient who exercises regularly may have lower body weight, and thus, require lower dosage of treatments, but also have an improved response to treatments. Hence, treatment dosage can be negatively correlated with treatment response, although administering lower dosage of treatments need not result in improved response. Hence, alternate methodologies need to be developed to compute the causal effect of treatments on health outcomes in the presence of unobserved confounders.

While it is impossible to precisely identify causal effects in the presence of unobserved confounders, it is possible to obtain bounds on the *causal query*, the causal effect of interest. There have been multiple such attempts to bound causal effects for small special graphs. Evans (2012) bound causal effects in the special case where any two observed variables are neither adjacent in the graph, nor share a latent parent. Richardson et al. (2014) bound the causal effect of a treatment on a parameter of interest by invoking additional (untestable) assumptions and assess how inference about the treatment effect changes as these assumptions are varied. Kilbertus et al. (2020) and Zhang and Bareinboim (2021) develop algorithms to compute causal bounds for extensions of the instrumental variable model in a continuous setting. Geiger and Meek (2013) bound causal effects in a model under specific parametric assumptions. Finkelstein and Shpitser (2020) develop a method for obtaining bounds on causal parameters using rules of probability and restrictions on counterfactuals implied by causal graphs.

While fewer in number, there have also been attempts to bound causal effects in large general graphs. Poderini et al. (2020) propose techniques to compute bounds in special large

graphs with multiple instruments and observed variables. Finkelstein et al. (2021) propose a method for partial identification in a class of measurement error models, and Zhang et al. (2022) and Duarte et al. (2021) propose a polynomial programming based approach to solve general causal inference problems, but their procedure is computationally intensive for large graphs.

In this work, we extend the class of large graphs for which causal effects can be efficiently bounded. In particular, we focus on a class of causal inference problems where causal bounds can be obtained using linear programming (LP) (Balke and Pearl, 1994; Zhang and Bareinboim, 2017; Pearl, 2009; Sjölander et al., 2014). Recently, Sachs et al. (2022) identified a large problem class for which LPs can be used to compute causal bounds, and have developed an algorithm for formulating the objective function and the constraints of the corresponding LP. This problem class is a generalization of the instrumental variable setting, and is thus widely applicable. However, as we describe later, the size of the LP is exponential in the number of edges in the causal graph, and therefore, the straightforward formulation of the LP can be tractably solved only for very small causal graphs. In this work, we show how to use the structure of the causal query and the underlying graph to significantly prune the LP, and as a consequence, significantly increase the size of the graphs for which the LP method remains tractable. This work is the full version of (Shridharan and Iyengar, 2022) and extends the pruning methodology to fractional linear programs that are used to compute bounds for causal inference problems with additional observations about the unit. Our main contributions are as follows:

(a) In Section 3 we show that the exponential number of variables in the LP used to compute causal bounds can be aggregated to reduce the number of variables by several orders of magnitude without impacting the quality of the bound – compare $|R|$ with $|H|$ in Table 1.

(b) Although we show the bounds can be computed by solving a much smaller LP, we get this computational advantage only if the pruned LP can be constructed efficiently. In Section 3 we show that the pruned LP can be constructed directly, i.e. without first constructing the original LP or iterating over its variables. These results critically leverage the structure of the LP corresponding to a causal inference problem. In particular, they leverage the fact that all possible functions from the parents $pa(V)$ to a variable V are admissible.

(c) In Section 4 we show that the structural results that help us construct the pruned LP lead to closed form bounds for a class of causal inference problems. This class of problems includes as a special case the well-studied setting in Figure 1 where multiple confounded treatments influence a outcome. Moreover, we are able to compute these bounds even when there are causal relationships between the treatments.

(d) In Section 5 we extend our pruning methodology to compute bounds for causal queries with additional observations about the unit. In this setting, the bounds are computed using fractional LPs. We show that the fractional LP can be converted into an LP, and then show how this LP can be pruned.

(e) In Section 6.3, we propose a simple greedy heuristic to compute approximate solutions for the pruned LPs when there are no additional observations. We show that this

heuristic allows us to compute approximate bounds for much larger graphs with very minimal degradation in performance.

The organization of the rest of this paper is as follows. In Section 2 we introduce our formalism. In Section 3 we introduce our main structural results for pruning the LPs. In Section 4 we show that the LP bounds can be computed in closed form for a large class of problems and also discuss an example of this class of problems. In Section 5 we show how to incorporate additional observations about the unit in the query to compute updated bounds. In Section 6 we report the results of numerical experiments validating and extending the methods proposed in Sections 3 and 5. In particular, we show the significant runtime improvement provided by our methods compared to benchmarks and extend all results in earlier sections to the finite data setting. In Section 6.3 we introduce our greedy heuristic and benchmark its performance. Section 7 discusses possible extensions.

2. Causal Inference Problems

Let G denote the causal graph. Let $V_1, \ldots, V_n$ denote the variables in G in topologically sorted order, and $N = \{1, \ldots, n\}$ denote the set of indices for the variables. We assume that each variable $V_i \in \{0, 1\}$. Later in this section we discuss why our proposed techniques automatically apply to the case where V_i takes discrete values. We use lower case letters for the values for the variables, and the notation $V_i = v_i$ denotes that the variable V_i takes the value $v_i \in \{0, 1\}$. For any subset $S \subseteq N$, we define $V_S := \{V_i : i \in S\}$, and the notation $V_S = v_S$ denotes the variable $V_i = v_i$, for all $i \in S$, for some $v \in \{0, 1\}^{|N|}$.

We consider a class of "partitioned" causal graphs with two sets of variables, V_A and V_B, where V_B topologically follow V_A (Sachs et al. (2020)). The V_A variables represent contextual or demographic variables (e.g. gender and age of a patient, past purchases, etc.) for a unit, and are always observed. For example, in Figure 1, $V_A = \{C_i : i = 1, 2\}$, the patient characteristics. The V_B variables can be observed, intervened upon or are the outcomes of interest in a query. In Figure 1 $V_B = \{Y\} \cup \{T_i : i = 1, \ldots, 5\}$.

Assumption 1 (Sachs et al. (2022)) *The index set N is partitioned into two sets $N = A \cup B$, where*

1. *V_B topologically follow V_A,*
2. *each variable in V_A has no parents, but is the parent of some variable in V_B.*
3. *V_B variables can have a common unobserved confounder U_B, and*
4. *no pair of variables (V_i, V_j), where $i \in A$ and $j \in B$, can share an unobserved confounder.*

For example, in the causal graph in Figure 2, $V_A = \{Z\}$ and $V_B = \{X, Y\}$.

We assume that the conditional probability distribution $p_{v_B.v_A} = \mathbb{P}(V_B = v_B | V_A = v_A)$ is known. In Section 6.2 we extend our results to the finite data setting where the estimate $\bar{p}_{v_B.v_A} \neq p_{v_B.v_A}$. Our goal is to compute bounds for counterfactual queries of the form $\mathcal{Q} = \mathbb{P}\big(V_O(V_I = q_I) = q_O | V_A = q_A\big)$, i.e. the probability of the outcome $V_O = q_O$ after an intervention $do(V_I = q_I)$ given contextual information for V_A. This counterfactual is specific to an individual realization of V_A, and computes the outcome of V_O if V_I was set to q_I. In order to simplify our results we impose a restriction on I that does not impact the query.

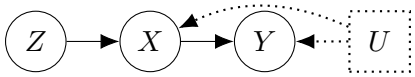


Figure 2: Causal Graph for Running Example with query $\mathcal{Q} = \mathbb{P}(Y(X = 1) = 1|Z = 1)$

Definition 2 (Critical variables for a query $\mathcal{Q}$) *Let $G^{do(V_I=q_I)}$ denote the mutilated graph after intervention $do(V_I = q_I)$, i.e. variables V_I no longer have any incoming arcs. Then the critical variables $V_{C(\mathcal{Q})}$ for the query $\mathcal{Q}$ denote the set of variables in $V_A \cup V_B$ that have a path to some variable in V_O in $G^{do(V_I=q_I)}$.*

Assumption 3 (Valid Query) *The query $\mathcal{Q} = \mathbb{P}\big(V_O(V_I = q_I) = q_O|V_A = q_A\big)$ satisfies the following conditions:*

(i) O, $I \subseteq B$, with $O \cap I = \emptyset$.

(ii) All variables $V_I \subseteq V_{C(\mathcal{Q})}$, i.e. all V_I variables are critical for the query.

Lemma 15 in Appendix A establishes that ii is without loss of generality since any $V_i \in V_I$ that is not critical can be removed from the query. Note that the class of problems defined in Sachs et al. (2022) allow the variables in V_A to have parents, but also require every variable in V_A that has a directed path to some variable in V_B in $G^{do(V_I=q_I)}$ to be intervened upon, and so these variables therefore cannot have parents in $G^{do(V_I=q_I)}$. Thus, the assumption in Sachs et al. (2022) is effectively equivalent to our assumption that variables in V_A do not have parents. Furthermore, we assume that every variable in V_A is the parent of some variable in V_B, and provides context in the query. Our assumption is more interpretable and broadly applicable, yet maintains the expressibility of the Sachs et al. (2022) formulation.

For graphs satisfying Assumption 1, the unobserved confounder U_B can potentially be very high dimensional with an unknown structure. We circumvent the difficulty of modeling U_B directly, by instead modeling the *impact* of the confounder on the relationships between the observed variables. For $j \in B$, let $pa(V_j)$ denote the parents of V_j in the causal graph. Then the variable $V_j = f_j(pa(V_j), U_B)$ for some unknown but fixed function $f_j : \{0,1\}^{|pa(V_j)|} \times \mathcal{U}_B \mapsto \{0,1\}$, where $\mathcal{U}_B$ denotes the domain of U_B. Therefore, the confounder U_B impacts the relationship between $pa(V_j)$ and V_j by selecting a function $f_j \in \mathcal{F}_j = \{f : f \text{ is a function from } pa(V_j) \mapsto V_j\}$. In the causal graph in Figure 2, for each fixed value for the unknown confounder U, the variable X is a function of Z; thus, U effectively selects one function from the set $\mathcal{F} = \{f \text{ is a function from } Z \mapsto X\}$. Similarly, U selects one function from the set $\mathcal{G} = \{g : g \text{ is a function from } X \mapsto Y\}$.

Since each variable $V_k \in pa(V_j)$ takes values in $\{0,1\}$ and $V_j \in \{0,1\}$, the cardinality of the set $|\mathcal{F}_j| = 2^{2^{|pa(V_j)|}}$. Therefore, the elements of $\mathcal{F}_j$ can be indexed by the set $R_{V_j} = \{1, \ldots, |\mathcal{F}_j|\}$. Let $f_j(\cdot, r_{V_j}) : pa(V_j) \mapsto V_j$ denote the r_{V_j}-th function in $\mathcal{F}_j$. Let the set $R = \prod_{j \in B} R_{V_j}$ index all possible mappings from $pa(V_j) \mapsto V_j$ for all $j \in B$. Thus, the response function variable $r = (r_{V_{|A|+1}}, \ldots r_{V_n}) \in R$ completely models the impact of U_B, i.e. the values of variables V_B is a deterministic function of V_A and r. For example, in the causal graph in Figure 2, it is easy to see that $|\mathcal{F}| = |\mathcal{G}| = 4$, and the set $R = \{(r_X, r_Y) = \{1, \ldots, 4\}^2\}$, where f_{r_X} denotes the r_X-th function from $\mathcal{F}$ and g_{r_Y} denotes

the r_Y-th function from $\mathcal{G}$. Note that the cardinality $|R| = \prod_{j \in B} 2^{2^{|pa(V_j)|}}$ is exponential in the number of arcs in the causal graph.

Note that although we work with causal graphs with binary variables in this paper, generalizing all subsequent results to categorical variables is straightforward. The critical property that we exploit is that the set of response function variables index possible mappings between variables. Therefore, our approach can be extended to general categorical variables by suitably defining response function variables. For example, suppose in Figure 2, $X, Y \in \{0, \ldots, m\}$. Then the response function variable $r_Y \in \{0, \ldots, m^m - 1\}$. All our results generalize to this more general setting.

The unknown distribution over the high dimensional U_B can be equivalently modeled via the distribution $\boldsymbol{q} \in \mathbb{R}_+^{|R|}$ over the set R. For $r \in R$, let $F_T(V_S = v_S, r)$ denote the value of $V_T \subseteq V_B$ when $V_S = v_S$ provided it is well defined. As discussed, setting $V_A = v_A$ and choosing $r \in R$ completely defines the values for V_B, i.e. $F_B(V_A = v_A, r)$ is well defined. Let

$$R_{v_B.v_A} = \{r : F_B(V_A = v_A, r) = v_B\}. \tag{1}$$

Hence, $p_{v_B.v_A} = \sum_{r \in R_{v_B.v_A}} q_r$. For example, in the causal graph in Figure 2, $R_{xy.z} = \{(r_X, r_Y) : f_{r_X}(z) = x, g_{r_Y}(x) = y\}$ denotes the set of r-values that map $z \mapsto (x, y)$. Hence, $\mathbb{P}(X = x, Y = y | Z = z) = \sum_{(r_X, r_Y) \in R_{xy.z}} q_{r_X r_Y}$.

The set

$$R_{\mathcal{Q}} = \big\{r \in R : F_O\big((V_A, V_I) = (q_A, q_I), r\big) = q_O\big\} \tag{2}$$

denotes the set of r values consistent with the query $\mathcal{Q} = \mathbb{P}(V_O(V_I = q_I) = q_O | V_A = q_A)$. Hence, $\mathbb{P}(V_O(V_I = q_I) = q_O | V_A = q_A) = \sum_{r \in R_{\mathcal{Q}}} q_r$. For the query $\mathcal{Q} = \mathbb{P}(Y(X = 1) = 1 | Z = 1)$ in the causal graph in Figure 2, the set $R_{\mathcal{Q}} = \{(r_X, r_Y) : g_{r_Y}(1) = 1\}$.

Then lower and upper bounds for the causal query can be obtained by solving the following pair of linear programs (Balke and Pearl (1994); Sachs et al. (2022)):

$$\begin{aligned} \alpha_L / \alpha_U = \min_q / \max_q \quad & \textstyle\sum_{r \in R_{\mathcal{Q}}} q_r \\ \text{s.t.} \quad & \textstyle\sum_{r \in R_{v_B.v_A}} q_r = p_{v_B.v_A}, \ \forall v_A, v_B, \\ & q \geq 0. \end{aligned} \tag{3}$$

Recall for $V_A = v_A$, $r \in R$ uniquely determines the value of V_B. Hence, for fixed v_A, $\cup_{v_B} R_{v_B.v_A}$ is a partition of R. Thus, the constraint $\sum_{r \in R} q_r = \sum_{v_B} \sum_{r \in R_{v_B.v_A}} q_r = \sum_{v_B} p_{v_B.v_A} = 1$ is implied by the other constraints in the LP, and therefore, is not explicitly added to the LP.

Note that our bounds are valid even if faithfulness assumptions are violated (Andersen, 2013). However, our results only leverage conditional independence relationships encoded in the graph, and not additional conditional independence relationships in the data. Hence, if the data displays additional independence relationships, our method is unable to leverage it to compute tighter bounds. Note also that we do not impose any additional constraints on the unobserved confounder.

3. Pruning the LP

In this section, we show how to reduce the size of the LPs (3) by aggregating variables. Let $h : V_A \to V_B$ denote any function $V_A \mapsto V_B$. We also refer to h as a *hyperarc* since it can be

interpreted as an arc in a hypergraph. We show how to reformulate LP (3) into another equivalent LP with variables indexed by hyperarcs, instead of response function variables. The number of possible hyperarcs is $(2^{|B|})^{2^{|A|}} \ll |R|$ (see Table 1). Hence, the new LP has at most $(2^{|B|})^{2^{|A|}}$ variables, and is thus exponentially smaller than LP (3). Furthermore, we show we only need to consider a smaller set of hyperarcs that are "valid" given the structure of the causal graph.

Recall that $F_B(V_A = v_A, r)$ denotes the value of V_B when $V_A = v_A$ and $r \in R$. Let

$$R_h = \Big\{ r \in R : F_B(V_A = v_A, r) = h(v_A),\ \forall v_A \in \{0,1\}^{|A|} \Big\} \tag{4}$$

denote the set of r values consistent with the hyperarc h, i.e. the set of r's that map $V_A = v_A$ to $V_B = h(v_A)$ for all inputs $v_A \in \{0,1\}^{|A|}$. For the graph in Figure 2,

$$R_h = \left\{ (r_X, r_Y) \in R : \begin{array}{l} \big(f_{r_X}(0), g_{r_Y}(f_{r_X}(0))\big) = h(0) \\ \big(f_{r_X}(1), g_{r_Y}(f_{r_X}(1))\big) = h(1) \end{array} \right\}$$

The causal graph structure implies that $R_h \neq \emptyset$ only for a subset of hyperarcs.

Definition 4 (Valid Hyperarc) *A hyperarc h is valid if $R_h \neq \emptyset$.*

In Section 3.1, we discuss why $R_h \neq \emptyset$ only for a subset of hyperarcs, and how to efficiently check the validity of a hyperarc. We denote the set of valid hyperarcs by H.

Next, we show how to write the LP in terms of variables $q_h = \sum_{r \in R_h} q_r$ corresponding to hyperarcs $h \in H$ by aggregating variables q_r for $r \in R_h$. In Lemma 14 in Appendix A we establish that $R = \cup_{h \in H} R_h$ is a partition of R. Therefore,

$$\sum_{r \in R_{v_B \cdot v_A}} q_r = \sum_{h \in H} \sum_{r \in R_h \cap R_{v_B \cdot v_A}} q_r = \sum_{\{h \in H : h(v_A) = v_B\}} \Big[\sum_{r \in R_h} q_r \Big] = \sum_{\{h \in H : h(v_A) = v_B\}} q_h.$$

Thus, the constraints in (3) can all be formulated in terms of the variables q_h corresponding to hyperarcs.

Next, consider the objective for the minimization LP:

$$\min \sum_{r \in R} \mathbf{1}\{r \in R_{\mathcal{Q}}\} q_r = \min \sum_{h \in H} \sum_{r \in R_h} \mathbf{1}\{r \in R_{\mathcal{Q}}\} q_r$$

From the definition of R_h, it follows that all q_r, $r \in R_h$, have a coefficient 1 in the same set of constraints, namely those indexed by $\{(v_B = h(v_A), v_A) : v_A \in \{0,1\}^{|A|}\}$. Hence, for any fixed value q_h for the variable corresponding to hyperarc h, any allocation in the set $\{[q_r]_{r \in R_h} : q_h = \sum_{r \in R_h} q_r, q_r \geq 0, r \in R_h\}$ is feasible. Hence, any optimal allocation satisfies

$$\min \Big\{ \sum_{r \in R_h} \mathbf{1}\{r \in R_{\mathcal{Q}}\} q_r : \sum_{r \in R_h} q_r = q_h, q_r \geq 0, \forall r \in R_h \Big\} = \Big(\min_{r \in R_h} \mathbf{1}\{r \in R_{\mathcal{Q}}\} \Big) q_h$$

Hence the lower bound LP can be reformulated as

$$\begin{array}{ll} \min_q & \sum_{h \in H} c_h^L q_h \\ \text{s.t.} & \sum_{h \in H : h(v_A) = v_B} q_h = p_{v_B \cdot v_A}, \quad \forall v_A, v_B, \\ & q \geq 0, \end{array} \tag{5}$$

where

$$c_h^L := \min_{r \in R_h} \mathbf{1}\{r \in R_{\mathcal{Q}}\} = \mathbf{1}\{R_h \subseteq R_{\mathcal{Q}}\}. \tag{6}$$

The objective for the upper bound is given by

$$\max \sum_{r \in R} \mathbf{1}\{r \in R_{\mathcal{Q}}\} q_r = \max \sum_{h \in H} \sum_{r \in R_h} \mathbf{1}\{r \in R_{\mathcal{Q}}\} q_r = \max \sum_{h \in H} \Big(\max_{r \in R_h} \mathbf{1}\{r \in R_{\mathcal{Q}}\} \Big) q_h,$$

where the second equality follows from an argument similar to the one used to establish the objective for the lower bound LP. Thus, the upper bound LP is given by

$$\begin{array}{ll} \max_q & \sum_{h \in H} c_h^U q_h \\ \text{s.t.} & \sum_{h \in H : h(v_A) = v_B} q_h = p_{v_B . v_A}, \quad \forall v_A, v_B \\ & q \geq 0, \end{array} \tag{7}$$

where

$$c_h^U = \max_{r \in R_h} \mathbf{1}\{r \in R_{\mathcal{Q}}\} = \mathbf{1}\{R_h \cap R_{\mathcal{Q}} \neq \emptyset\}. \tag{8}$$

Both reformulations have *exponentially* fewer variables since $|H| \ll |R|$; however, they are useful only if the set of valid hyperarcs H and the corresponding coefficients $\mathbf{1}\{R_h \subseteq R_{\mathcal{Q}}\}$ and $\mathbf{1}\{R_h \cap R_{\mathcal{Q}} \neq \emptyset\}$ can be efficiently computed, i.e. in particular, without formulating the original LPs or iterating over R. In Section 3.1 we describe how to efficiently check the validity of a hyperarc and efficiently compute H, and in Section 3.2 (resp. Section 3.3) we show how to efficiently compute c_h^L (resp. c_h^U). Finally, in Section 3.4, we describe our procedure that uses results established in Sections 3.1, 3.2 and 3.3 to efficiently construct pruned LPs (5) and (7) without formulating the original LPs or iterating over R.

3.1 Characterizing Valid Hyperarcs

We now discuss why $R_h \neq \emptyset$ only for a subset of hyperarcs. We also show how to efficiently check the validity of hyperarc h without enumerating all values in the set R to check if there exists $r \in R$ such that $F_B(V_A = v_A, r) = h(v_A)$ for all $v_A \in \{0,1\}^{|A|}$ i.e. $R_h \neq \emptyset$. Instead, we show that the outputs of the function h alone are sufficient to determine its validity. We motivate the main result of this section by first considering the simple causal graph in Figure 2. Consider a hyperarc h with $h(0) = (x_0, y_0)$ and $h(1) = (x_1, y_1)$ for $x_i, y_i \in \{0,1\}$, $i = 0, 1$. Note that for any choice of x_i and y_i, $i = 0, 1$, h is a hyperarc from $Z \mapsto (X, Y)$. For h to be a valid hyperarc, it should be of the form $h(z) = (f_h(z), g_h(f_h(z)))$ for some $f_h \in \mathcal{F}$ and $g_h \in \mathcal{G}$. The functions (if they exist) must satisfy:

$$f_h(z) = \begin{cases} x_0 & \text{if } z = 0 \\ x_1 & \text{if } z = 1 \end{cases} \qquad g_h(x) = \begin{cases} y_0 & \text{if } x = x_0 \\ y_1 & \text{if } x = x_1 \end{cases}$$

Clearly, f_h is well defined for any choice of x_i, $i = 0, 1$. However, there exists g_h satisfying the conditions above if, and only if, $y_0 = y_1$ whenever $x_0 = x_1$. Hence, $h \notin H$ if, and only if, $x_0 = x_1$ but $y_0 \neq y_1$. For example, consider the hyperarc h_1 with $h_1(0) = (0, 0)$ and $h_1(1) = (0, 0)$. To check if h_1 is valid, we need to check if there exists well defined functions

$f_{h_1} : Z \mapsto X$ and $g_{h_1} : X \mapsto Y$ which satisfy:

$$f_{h_1}(z) = \begin{cases} 0 & \text{if } z = 0 \\ 0 & \text{if } z = 1 \end{cases} \qquad g_{h_1}(x) = 0, \quad \text{if } x = 0$$

Clearly, f_{h_1} is a well defined function. Any function $g : X \mapsto Y$ in $\mathcal{G}$ with $g(0) = 0$ satisfies the conditions for g_{h_1}. Hence, $h_1 \in H$.

On the other hand, consider the hyperarc h_2 with $h_2(0) = (0, 0)$ and $h_2(1) = (0, 1)$. The hyperarc h_2 is valid if there exists a well defined function $g_{h_2} : X \mapsto Y$ which satisfies:

$$g_{h_2}(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x = 0 \end{cases}$$

Clearly, there cannot be such a function, and so $h_2 \notin H$.

Note that we did not have to iterate over the set R to check if a hyperarc h is valid. Instead, we recognize that a hyperarc h only partially specifies a function mapping from $pa(V_j)$ to V_j. Therefore, in order to check the validity of h, we only need to check if there exists some binary function $pa(V_j) \to V_j$ which satisfies this partial specification. Hence, the outputs of the function h alone are sufficient to determine its validity. The following theorem generalizes this observation to general causal graphs.

Theorem 5 (Valid hyperarcs) *Let $a_A \in \{0, 1\}^{|A|}$ denote the values set for the variables V_A, and let $a_B = h(a_A) \in \{0, 1\}^{|B|}$ denote the values for the variables V_B when the hyperarc h is evaluated at a_A. A hyperarc h is valid if, and only if, for all $\big(a_A, a_B = h(a_A)\big), \big(b_A, b_B = h(b_A)\big) \in \{0, 1\}^{|N|}$ and $j \in B$,*

$$a_{P_j} = b_{P_j} \implies a_j = b_j, \tag{9}$$

where $P_j \subseteq N$ denote the indices of $pa(V_j)$.

Proof It is clear that if a hyperarc is valid, then it is satisfies (9).

Suppose a hyperarc satisfies (9). Then we are given a partial specification for a *function* from $pa(V_j) \to V_j$, i.e. the same input values are always mapped to the same output value; however, the output is only specified for possibly a subset of input values. For each $V_j \in V_B$, r_{V_j} indexes the set of all possible functions $pa(V_j) \to V_j$; therefore, for every node j there exists a binary function $pa(V_j) \to V_j$ which satisfies the partial specification given by the hyperarc h. Thus, $R_h \neq \emptyset$, or equivalently, h is valid. ■

Theorem 5 implies that we only have to search through the set of possible hyperarcs from $V_A \mapsto V_B$ of cardinality $2^{|B|2^{|A|}}$ for hyperarcs which satisfy (9) to identify the set H of valid hyperarcs. The first two columns of Table 1 compare $|R|$ with the maximum possible number of hyperarcs $2^{|B|2^{|A|}}$ for seven different causal inference problems (details in Appendix). Note that the reduction in size can be several orders of magnitude, and it increases with the complexity of the causal graph, see e.g. Examples D and E. Thus, there is a very significant reduction in size even if all hyperarcs are valid. The last column in Table 1 lists $|H|$. Considering only the valid hyperarcs further decreases the size of the LP by at least 1 order of magnitude, and sometimes more. The LPs corresponding to Examples B

Graph	$\lvert R\rvert$	$2^{\lvert B\rvert 2^{\lvert A\rvert}}$	$\lvert H\rvert$
Ex A	1.3×10^8	1.0×10^6	2.3×10^3
Ex B	4.2×10^6	1.0×10^6	7.1×10^4
Ex C	4.2×10^6	1.0×10^6	4.4×10^4
Ex D	6.3×10^{57}	1.7×10^7	9.4×10^6
Ex E	3.2×10^{32}	1.7×10^7	9.4×10^6
Ex F	1.8×10^{13}	6.5×10^4	5.8×10^4
Ex G	3.0×10^{23}	1.0×10^6	9.2×10^3

Table 1: The naive LP for computing causal bounds has $|R|$ variables, where $|R|$ denotes the cardinality of the set of all possible values for the response function variables. The number of variables drops to $2^{|B|2^{|A|}}$ when the LP is formulated in terms of hyperarcs, and the number of variables can be further reduced to $|H|$, the cardinality of the set of *valid* hyperarcs. Note that $|R| \gg 2^{|B|2^{|A|}} \gg |H|$. See Section 3 for details.

and C can be solved without pruning; however, the LPs corresponding to Examples A, F and G can only be solved after pruning the problem, and the LPs for Examples D and E are too large even after pruning. In Section 6.3 we propose a greedy heuristic to compute bounds for these problems. Next, we show how to efficiently compute $c_h^L = \mathbf{1}\{R_h \subseteq R_\mathcal{Q}\}$ and $c_h^U = \mathbf{1}\{R_h \cap R_\mathcal{Q} \neq \emptyset\}$.

3.2 Efficiently computing $c_h^L = \mathbf{1}\{R_h \subseteq R_\mathcal{Q}\}$

We show how to check if $R_h \subseteq R_\mathcal{Q}$ efficiently i.e. without iterating over the set R_h and checking if each value lies in $R_\mathcal{Q}$. Instead, we show that the outputs of h alone are sufficient to determine if $R_h \subseteq R_\mathcal{Q}$. As before, we illustrate the main ideas using the graph in Figure 2 with query $\mathcal{Q} = \mathbb{P}(Y(X=1)=1|Z=1)$, and then prove them.

Theorem 6 *Suppose the causal graph satisfies Assumption 1 and the query $\mathcal{Q}$ satisfies Assumption 3. Then $R_h \subseteq R_\mathcal{Q}$ if, and only if, there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, $v_I = q_I$ and $v_O = q_O$.*

Consider the query $\mathcal{Q} = \mathbb{P}(Y(X=1)=1|Z=1)$ in the causal graph in Figure 2. Here $V_A = \{Z\}$, $V_B = \{X, Y\}$, $V_I = \{X\}$ and $V_O = \{Y\}$. For a hyperarc $h : Z \mapsto (X, Y)$, Theorem 6 implies

$$R_h \subseteq R_\mathcal{Q} \iff h(z) = (1,1) \text{ for some } z \in \{0,1\}. \tag{10}$$

Hence, Theorem 6 implies that we can efficiently compute $c_h^L = \mathbf{1}\{R_h \subseteq R_\mathcal{Q}\}$ for a hyperarc h by considering only the outputs of the hyperarc, instead of R_h.

Proof Suppose there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, $v_I = q_I$ and $v_O = q_O$. Then, every $r \in R_h$ maps $(V_{A\cap C(\mathcal{Q})}, V_I) = (q_{A\cap C(\mathcal{Q})}, q_I)$ to $V_O = q_O$, and thus, $r \in R_\mathcal{Q}$ i.e. $R_h \subseteq R_\mathcal{Q}$. To establish the opposite direction, suppose $R_h \subseteq R_\mathcal{Q}$, but there does not exist $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, $v_I = q_I$ and $v_O = q_O$. Equivalently, for all $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, we have either $v_I \neq q_I$ or $v_O \neq q_O$. We consider these two cases separately.

(a) For all $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, we have $v_I \neq q_I$: Since R indexes all possible functions on the causal graph, there exists $r \in R$ such that:

- r maps $V_A = v_A$ to $V_B = h(v_A)$ for all v_A, i.e. $r \in R_h$.
- r maps $V_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, and $V_I = q_I \neq v_I$ to $V_O \neq q_O$.

Hence $R_h \nsubseteq R_{\mathcal{Q}}$, a contradiction.

(b) There exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$ and $v_I = q_I$ but $v_O \neq q_O$: In this case, every $r \in R_h$ maps $(V_{A\cap C(\mathcal{Q})}, V_I) = (q_{A\cap C(\mathcal{Q})}, q_I)$ to $V_O \neq q_O$, and therefore, $r \notin R_{\mathcal{Q}}$. Hence $R_h \nsubseteq R_{\mathcal{Q}}$.

■

3.3 Efficiently computing $c_h^U = \mathbf{1}\{R_h \cap R_{\mathcal{Q}} \neq \emptyset\}$

We now show how to efficiently check if $R_h \cap R_{\mathcal{Q}} = \emptyset$. As in the case with c_h^L, we show that the condition $R_h \cap R_{\mathcal{Q}} = \emptyset$ can be checked without iterating over the set R_h. Instead, the outputs of h alone are sufficient to determine if $R_h \cap R_{\mathcal{Q}} = \emptyset$.

Theorem 7 *Suppose the causal graph satisfies Assumption 1 and the query satisfies Assumption 3. Then $R_h \cap R_{\mathcal{Q}} = \emptyset$ if, and only if, there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, $v_I = q_I$ and $v_O \neq q_O$.*

Again, consider the query $\mathcal{Q} = \mathbb{P}(Y(X=1)=1|Z=1)$ in the causal graph in Figure 2. For a hyperarc $h : Z \mapsto (X, Y)$, Theorem 7 implies

$$R_h \cap R_{\mathcal{Q}} = \emptyset \iff h(z) = (1,0) \text{ for some } z \in \{0,1\}. \tag{11}$$

Hence, Theorem 7 implies that we can efficiently compute $c_h^U = \mathbf{1}\{R_h \cap R_{\mathcal{Q}} \neq \emptyset\}$ for a hyperarc h by considering only the outputs of the hyperarc, instead of R_h.

Proof Suppose there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, $v_I = q_I$ and $v_O \neq q_O$. Then, every $r \in R_h$ maps $(V_{A\cap C(\mathcal{Q})}, V_I) = (q_{A\cap C(\mathcal{Q})}, q_I)$ to $V_O \neq q_O$, and thus, $r \notin R_{\mathcal{Q}}$ i.e. $R_h \cap R_{\mathcal{Q}} = \emptyset$. To establish the opposite direction, suppose $R_h \cap R_{\mathcal{Q}} = \emptyset$, but there does not exist $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, $v_I = q_I$ and $v_O \neq q_O$. Equivalently, for all $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, we have either $v_I \neq q_I$ or $v_O = q_O$. We consider these two cases separately.

(a) For all $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, we have $v_I \neq q_I$: Since R indexes all possible functions on the causal graph, there exists $r \in R$ such that:

- r maps $V_A = v_A$ to $V_B = h(v_A)$ for all v_A, i.e. $r \in R_h$.
- r maps $V_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, and $V_I = q_I \neq v_I$ to $V_O = q_O$.

Hence $R_h \cap R_{\mathcal{Q}} \neq \emptyset$, a contradiction.

Algorithm 1 Procedure to efficiently construct LPs (5), (7)

Input: (i) causal graph G, (ii) query $\mathcal{Q} = \mathbb{P}(V_O(V_I = q_I) = q_O | V_A = q_A)$, (iii) conditional probability distribution $p_{v_B.v_A} = \mathbb{P}(V_B = v_B | V_A = v_A)$, for all v_A, v_B.
Output: Pruned LPs (5) and (7)
$H \leftarrow \emptyset$
for $h : V_A \to V_B$ **do**
  if *h is valid (Theorem 5)* **then**
    $H \leftarrow H \cup \{h\}$
    Compute c_h^L using Theorem 6
    Compute c_h^U using Theorem 7
return *LPs* (5) *and* (7) *constructed using* (H, c^L, c^U)

(b) There exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{A \cap C(\mathcal{Q})} = q_{A \cap C(\mathcal{Q})}$ and $v_I = q_I$ but $v_O = q_O$: In this case, every $r \in R_h$ maps $(V_{A \cap C(\mathcal{Q})}, V_I) = (q_{A \cap C(\mathcal{Q})}, q_I)$ to $V_O = q_O$, and therefore, $r \in R_{\mathcal{Q}}$. Hence $R_h \cap R_{\mathcal{Q}} \neq \emptyset$.

■

3.4 Computing the Pruned LPs

Algorithm 1 describes a procedure that uses results established in Sections 3.1, 3.2 and 3.3 to efficiently construct pruned LPs (5) and (7) without formulating the original LPs or iterating over R.

4. Bounds in Closed Form

Now we show that if $A \subseteq \mathcal{C}(\mathcal{Q})$, i.e. all V_A variables are critical for the query, the bounds can be computed in closed form by simply adding appropriate conditional probabilities in the input data. This is in contrast to the closed form bounds in Balke and Pearl (1994) and Sachs et al. (2022) that are computed by enumerating vertices of the constraint polytope. These bounds cannot be computed for large graphs since the total number of vertices grows exponentially in the size of the associated LP, which itself is large. On the other hand, our bounds, which involve simply adding probabilities in the input data, scale significantly better to larger problems.

We motivate the main ideas of the result using the causal graph in Figure 3 and query $\mathcal{Q} = \mathbb{P}(Y(X = 1) = 1 | Z = 1)$. Note that Z is critical for this query in this graph due to the edge $Z \to Y$. The results in Section 3.2 imply that $c_h^L = 1$ if, and only if, $h(1) = (1, 1)$. Thus, it follows that the objective of the pruned LP for the lower bound can be written as

$$\begin{aligned} \sum_{\{h \in H : c_h^L = 1\}} q_h &= \sum_{\{h \in H : h(1) = (1,1)\}} q_h \\ &= p_{11.1} \end{aligned} \tag{12}$$

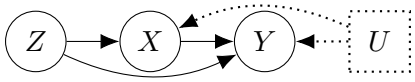


Figure 3: Causal Graph for Query $\mathcal{Q}_1$

where (12) follows from the constraints of the pruned LP (5). Hence, the lower bound can be computed in closed form.

On the other hand, consider the (original) causal graph in Figure 2 and the query $\mathcal{Q} = \mathbb{P}(Y(X = 1) = 1|Z = 1)$. Now Z is no longer critical for the query in this graph. From (10) we have that $c_h^L = 1$ if, and only if, there exists $z \in \{0, 1\}$ such that $h(z) = (1, 1)$. The objective of the pruned LP is:

$$\begin{aligned}\sum_{\{h\in H: c_h^L=1\}} q_h &= \sum_{\{h\in H: \exists z\in\{0,1\}, h(z)=(1,1)\}} q_h \\ &\neq \sum_{h\in H: h(0)=(1,1)} q_h + \sum_{h\in H: h(1)=(1,1)} q_h,\end{aligned}$$

since $\{h \in H : h(0) = (1, 1)\} \cap \{h \in H : h(1) = (1, 1)\} \neq \emptyset$. Hence, we cannot compute the lower bound in closed form for the query. It appears that we need the input variable to the hyperarc h to be critical for the query in order to compute closed form bounds, which is formalized by the condition $A \subseteq C(\mathcal{Q})$.

Theorem 8 (Closed Form Bounds for Special Class of Problems) *Suppose $A \subseteq C(\mathcal{Q})$. Then the optimal values of LPs* (5) *and* (7) *are given by*

$$\begin{aligned}\alpha_L &= \sum_{\{v_B: v_I=q_I, v_O=q_O\}} p_{v_B.q_A}, \\ \alpha_U &= 1 - \sum_{\{v_B: v_I=q_I, v_O\neq q_O\}} p_{v_B.q_A}.\end{aligned}$$

Proof Theorem 6 implies $R_h \subseteq R_{\mathcal{Q}}$ if, and only if, there exists $v = (v_A, v_B)$ such that $h(v_A) = v_B$, $v_{A\cap C(\mathcal{Q})} = q_{A\cap C(\mathcal{Q})}$, $v_I = q_I$, and $v_O = q_O$. Since $A \subseteq C(\mathcal{Q})$, it follows that $A \cap C(\mathcal{Q}) = A$, and therefore, for a hyperarc h, the set $R_h \subseteq R_{\mathcal{Q}}$ if, and only if, there exists v_B such that $h(q_A) = v_B$, $v_I = q_I$, and $v_O = q_O$. Thus, it follows that

$$\begin{aligned}\alpha_L &= \sum_{\{h\in H: c_h^L=1\}} q_h \\ &= \sum_{\{h\in H:\ R_h \subseteq R_{\mathcal{Q}}\}} q_h \\ &= \sum_{\{v_B: v_I=q_I, v_O=q_O\}} \sum_{\{h\in H: h(q_A)=v_B\}} q_h \qquad (13) \\ &= \sum_{\{v_B: v_I=q_I, v_O=q_O\}} p_{v_B.q_A} \qquad (14)\end{aligned}$$

where (13) follows from the discussion above and (14) from the constraints of the pruned LP.

Theorem 7 implies that $R_h \cap R_{\mathcal{Q}} = \emptyset$ if, and only if, there exists $v = (v_A, v_B)$ such that $h(v_A) = v_B$, $v_{A \cap C(\mathcal{Q})} = q_{A \cap C(\mathcal{Q})}$, $v_I = q_I$, and $v_O \neq q_O$. Since $A \subseteq C(\mathcal{Q})$, it follows that $R_h \cap R_{\mathcal{Q}} = \emptyset$ if, and only if, there exists v_B such that $h(q_A) = v_B$, $v_I = q_I$, and $v_O \neq q_O$. Thus, it follows that

$$\begin{aligned}
\sum_{\{h \in H : c_h^U = 0\}} q_h &= \sum_{\{h \in H:\ R_h \cap R_{\mathcal{Q}} = \emptyset\}} q_h \\
&= \sum_{\{v_B : v_I = q_I, v_O \neq q_O\}} \ \sum_{\{h \in H : h(q_A) = v_B\}} q_h \qquad (15) \\
&= \sum_{\{v_B : v_I = q_I, v_O \neq q_O\}} p_{v_B . q_A} \qquad (16)
\end{aligned}$$

where (15) follows from the discussion above and (16) from the constraints of the pruned LP. The result follows by the fact that $\alpha_U = \sum_{\{h \in H : c_h^U = 1\}} q_h = 1 - \sum_{\{h \in H : c_h^U = 0\}} q_h$. ■

An important example of a class of problems where $A \subseteq C(\mathcal{Q})$ is the well-studied setting where multiple confounded treatments influence one outcome (Ranganath and Perotte, 2019; Janzing and Schölkopf, 2018; D'Amour, 2019; Tran and Blei, 2017). An example from this class of problems was discussed in Section 1, where the treatments are medications and procedures, and the outcome is the health outcome of the patient.

Wang and Blei (2019a, 2021) introduced the *deconfounder* as a method to predict the expected value of the outcome variable under treatment interventions in this setting. One of the limitations of the deconfounder approach is that it cannot be applied in a setting where the treatment variables have causal relationships between them (Ogburn et al., 2019; Wang and Blei, 2019b; Imai and Jiang, 2019). For example, in our context, the side effects of one treatment can influence the prescription of another treatment (Ogburn et al., 2019), as implied by arrows $T_1 \to T_3$, $T_2 \to T_3$, $T_3 \to T_4$ and $T_3 \to T_5$ in Figure 4. The deconfounder cannot be used for inference in this setting. However, since the entire set $V_A = \{C_1, C_2\}$ is critical for the query, Theorem 8 can be used to compute bounds for the query $\mathbb{E}[Y(\mathbf{T} = t) | C_1 = c_1, C_2 = c_2]$ in closed form. In particular,

$$\begin{aligned}
\alpha_L &= \mathbb{P}(\mathbf{T} = t, Y = 1 | C_1 = c_1, C_2 = c_2) \\
\alpha_U &= 1 - \mathbb{P}(\mathbf{T} = t, Y = 0 | C_1 = c_1, C_2 = c_2)
\end{aligned}$$

5. Bounds with Additional Observations

So far, we have allowed only observations of V_A variables to provide context in a query. In this section, we show how to incorporate additional observed variables from V_B in the query. We show how to efficiently compute bounds for the counterfactual query $\mathcal{Q}_W = \mathbb{P}\big(V_O(V_I = q_I) = q_O | V_A = q_A, V_W = q_W\big)$ where $V_W \subseteq V_B$.

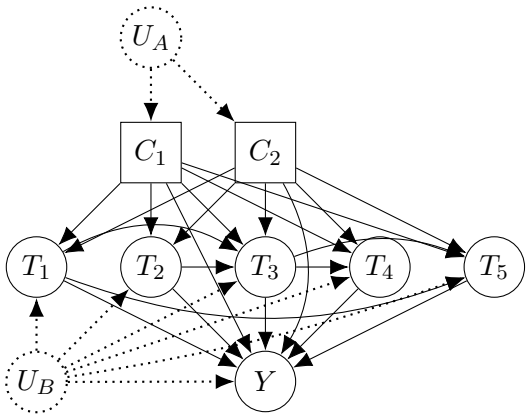


Figure 4: Example of G where $V_A = \{C_1, C_2\}$ and $V_B = \{T_1, T_2, T_3, T_4, T_5, Y\}$

Define

$$R_{\mathcal{W}} = \{r : F_W(V_A = q_A, r) = q_W\}.$$

to be the set of r values that are consistent with the observation $(V_A, V_W) = (q_A, q_W)$. Note that since V_W topologically follow V_A, the function $F_W(V_A = q_A, r)$, and thus the set $R_{\mathcal{W}}$, are well defined.

Let $\mathcal{Q} = \mathbb{P}(V_O(V_I = q_I) = q_O | V_A = q_A)$. The revised objective is then given by:

$$\begin{aligned}
&\mathbb{P}(V_O(V_I = q_I) = q_O | V_A = q_A, V_W = q_W) \\
&\qquad = \frac{\mathbb{P}(\{V_O(V_I = q_I, V_A = q_A) = q_O\} \cap \{V_W(V_A = q_A) = q_W\})}{\mathbb{P}(V_W(V_A = q_A) = q_W)} && (17) \\
&\qquad = \frac{\sum_{r \in R_{\mathcal{Q}} \cap R_{\mathcal{W}}} q_r}{\sum_{r \in R_{\mathcal{W}}} q_r} && (18)
\end{aligned}$$

where (17) follows from Bayes' rule and (18) follows from the definitions of $R_{\mathcal{Q}}$ and $R_{\mathcal{W}}$. Note that our objective is now a *fractional linear* expression in q_r, while our constraints remain the same. However, since the fractional objective is always non-negative and bounded, the fractional LPs can be reformulated into the following LPs (Charnes and Cooper (1962)):

$$\begin{array}{rl}
\min_{\alpha,q} / \max_{\alpha,q} & \sum_{r \in R_{\mathcal{Q}} \cap R_{\mathcal{W}}} q_r \\
\text{s.t.} & \sum_{r \in R_{v_B \cdot v_A}} q_r - \alpha p_{v_B \cdot v_A} = 0, \quad \forall (v_A, v_B) \in \{0,1\}^{|A|} \times \{0,1\}^{|B|}, \\
& \sum_{r \in R_{\mathcal{W}}} q_r = 1, \\
& \alpha, q \geq 0,
\end{array} \tag{19}$$

For the causal graph in Figure 2 and counterfactual $\mathbb{P}(Y(X = 1) = 1 | X = 0, Z = 1)$, the set $R_{\mathcal{W}} = \{(r_X, r_Y) : f_{r_X}(1) = 0\}$, and the reformulated LPs are given by:

$$\begin{array}{rl}
\min_{\alpha,q} / \max_{\alpha,q} & \sum_{r \in R_{\mathcal{Q}} \cap R_{\mathcal{W}}} q_r \\
\text{s.t.} & \sum_{r \in R_{xy \cdot z}} q_r - \alpha p_{xy \cdot z} = 0, \quad \forall (x, y, z) \in \{0,1\}^3 \\
& \sum_{r \in R_{\mathcal{W}}} q_r = 1, \\
& \alpha, q \geq 0,
\end{array} \tag{20}$$

where $\mathcal{Q} = \mathbb{P}(Y(X=1)=1|Z=1)$. Next, we show that the LP (20) can be reformulated in terms of variables $q_h = \sum_{r \in R_h} q_r$ indexed by valid hyperarcs $h \in H$. For that we need the following result.

Theorem 9 *For every valid hyperarc $h \in H$, either $R_h \subseteq R_{\mathcal{W}}$ or $R_h \cap R_{\mathcal{W}} = \emptyset$.*

Proof Fix $h \in H$. Suppose there exists $v \in \{0,1\}^N$ such that $h(v_A) = v_B$, $v_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ and $v_W = q_W$. Then from Theorem 11a, we have $R_h \subseteq R_{\mathcal{W}}$. Instead, suppose for every $v \in \{0,1\}^N$ such that $h(v_A) = v_B$ and $v_{C(\mathcal{W})} = q_{C(\mathcal{W})}$, we have $v_W \neq q_W$. Then, Theorem 11b implies that $R_h \cap R_{\mathcal{W}} = \emptyset$. Hence, either $R_h \subseteq R_{\mathcal{W}}$ or $R_h \cap R_{\mathcal{W}} = \emptyset$. ■

Since $R = \bigcup_h R_h$ is a partition of R (see Lemma 14), it follows that the constraint

$$\sum_{r \in R_{\mathcal{W}}} q_r = \sum_{h \in H} \Big[\sum_{r \in R_{\mathcal{W}} \cap R_h} q_r \Big] = \sum_{h \in H: R_h \subseteq R_{\mathcal{W}}} \Big[\sum_{r \in R_h} q_r \Big] = \sum_{h \in H: R_h \subseteq R_{\mathcal{W}}} q_h$$

where the second equality follows from Theorem 9. The results in Section 3 imply that for any $(v_A, v_B) \in \{0,1\}^{|A|} \times \{0,1\}^{|B|}$

$$\sum_{r \in R_{v_B . v_A}} q_r = \sum_{\{h \in H: h(v_A) = v_B\}} \sum_{r \in R_h} q_r = \sum_{\{h \in H: h(v_A) = v_B\}} q_h.$$

Thus, the constraints can be written in terms of q_h. An argument identical to the one in Section 3 implies that the objective coefficient of q_h in the minimization LP is given by

$$d_h^L := \mathbf{1}\{R_h \subseteq R_{\mathcal{Q}} \cap R_{\mathcal{W}}\} = \mathbf{1}\{R_h \subseteq R_{\mathcal{W}}\}\mathbf{1}\{R_h \subseteq R_{\mathcal{Q}}\}$$

Thus, the lower bound LP can be reformulated as follows.

$$\begin{array}{rl} \min_q & \sum_{h \in H} d_h^L q_h \\ \text{s.t.} & \sum_{h \in H: h(v_A) = v_B} q_h = \alpha p_{v_B . v_A}, \quad \forall v_A, v_B, \\ & \sum_{h \in H: R_h \subseteq R_{\mathcal{W}}} q_h = 1, \\ & q, \alpha \geq 0, \end{array} \tag{21}$$

Similarly, the upper bound LP can be reformulated as:

$$\begin{array}{rl} \max_q & \sum_{h \in H} d_h^U q_h \\ \text{s.t.} & \sum_{h \in H: h(v_A) = v_B} q_h = \alpha p_{v_B . v_A}, \quad \forall v_A, v_B, \\ & \sum_{h \in H: R_h \subseteq R_{\mathcal{W}}} q_h = 1, \\ & q, \alpha \geq 0, \end{array} \tag{22}$$

where

$$d_h^U := \mathbf{1}\{R_h \cap (R_{\mathcal{W}} \cap R_{\mathcal{Q}}) \neq \emptyset\} = 1 - \mathbf{1}\{R_h \cap (R_{\mathcal{W}} \cap R_{\mathcal{Q}}) = \emptyset\}. \tag{23}$$

The new LPs in terms of the q_h variables have *exponentially* fewer variables since $|H| \ll |R|$; however, they are useful only if, for each hyperarc $h \in H$, d_h^L, d_h^U, and $\mathbf{1}\{R_h \subseteq R_{\mathcal{W}}\}$ can be efficiently computed.

5.1 Efficiently computing d_h^L, d_h^U, and $\mathbf{1}\{R_h \subseteq R_{\mathcal{W}}\}$

We begin by defining the set of critical variables for the observation V_W.

Definition 10 (Critical variables for observation $\{V_W = q_W\}$) *The set $V_{C(\mathcal{W})} \subseteq V_A$ of critical variables for the observation $\{V_W = q_W\}$ is defined as the set of variables in V_A that have a path in the graph G to some variable in V_W.*

Next, we define conditions under which $R_h \subseteq R_{\mathcal{W}}$, $R_h \cap R_{\mathcal{W}} = \emptyset$, and $R_h \cap (R_{\mathcal{Q}} \cap R_{\mathcal{W}}) = \emptyset$.

Theorem 11 *Let $V_{C(\mathcal{W})} \subseteq V_A$ denote the set of critical variables for the observation $\{V_W = q_W\}$. Then the following results hold.*

(a) $R_h \subseteq R_{\mathcal{W}}$ if, and only if, there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ and $v_W = q_W$.

(b) $R_h \cap R_{\mathcal{W}} = \emptyset$ if, and only if, there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ and $v_W \neq q_W$.

Proof Suppose there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ and $v_W = q_W$. Then every $r \in R_h$ maps $V_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ to $V_W = q_W$, and thus, $r \in R_{\mathcal{W}}$ i.e. $R_h \subseteq R_{\mathcal{W}}$. To establish the reverse direction, suppose $R_h \subseteq R_{\mathcal{W}}$, but there does not exist $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ and $v_W = q_W$. Equivalently, for all $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and $v_{C(\mathcal{W})} = q_{C(\mathcal{W})}$, we have $v_W \neq q_W$. In this case, every $r \in R_h$ maps $V_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ to $V_W \neq q_W$, and therefore, $r \notin R_{\mathcal{W}}$. Hence $R_h \nsubseteq R_{\mathcal{W}}$.

Suppose there exists $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ and $v_W \neq q_W$. Then, every $r \in R_h$ maps $V_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ to $V_W \neq q_W$, and thus, $r \notin R_{\mathcal{W}}$ i.e. $R_h \cap R_{\mathcal{W}} = \emptyset$. To establish the opposite direction, suppose $R_h \cap R_{\mathcal{W}} = \emptyset$, but there does not exist $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$, $v_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ and $v_W \neq q_W$. Equivalently, for all $v \in \{0,1\}^{|N|}$ such that $h(v_A) = v_B$ and $v_{C(\mathcal{W})} = q_{C(\mathcal{W})}$, we have $v_W = q_W$. In this case, every $r \in R_h$ maps $V_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ to $V_W = q_W$, and therefore, $r \in R_{\mathcal{W}}$. Hence $R_h \cap R_{\mathcal{W}} \neq \emptyset$. ■

If $R_{\mathcal{Q}} \cap R_{\mathcal{W}} = \emptyset$, the probability of both the observation and the query would be 0. Hence, the observation would invalidate the query, and there would be no need for bounds. Therefore, we assume that $R_{\mathcal{Q}} \cap R_{\mathcal{W}} \neq \emptyset$, and in that we have the following result.

Theorem 12 *Suppose $R_{\mathcal{Q}} \cap R_{\mathcal{W}} \neq \emptyset$. Then $R_h \cap (R_{\mathcal{Q}} \cap R_{\mathcal{W}}) = \emptyset$ if, and only if, either $R_h \cap R_{\mathcal{Q}} = \emptyset$ or $R_h \cap R_{\mathcal{W}} = \emptyset$.*

Proof Clearly, if either $R_h \cap R_{\mathcal{Q}} = \emptyset$ or $R_h \cap R_{\mathcal{W}} = \emptyset$, it follows that $R_h \cap (R_{\mathcal{Q}} \cap R_{\mathcal{W}}) = \emptyset$.

Suppose $R_h \cap (R_{\mathcal{Q}} \cap R_{\mathcal{W}}) = \emptyset$, but $R_h \cap R_{\mathcal{Q}} \neq \emptyset$ and $R_h \cap R_{\mathcal{W}} \neq \emptyset$. By Theorem 7 and Theorem b, there cannot exist either $v^{\mathcal{W}} \in \{0,1\}^{|N|}$ such that $h(v_A^{\mathcal{W}}) = v_B^{\mathcal{W}}$, $v_{C(\mathcal{W})}^{\mathcal{W}} = q_{C(\mathcal{W})}$ and $v_W^{\mathcal{W}} \neq q_W$, or $v^{\mathcal{Q}} \in \{0,1\}^{|N|}$ such that $h(v_A^{\mathcal{Q}}) = v_B^{\mathcal{Q}}$, $v_{A \cap C(\mathcal{Q})}^{\mathcal{Q}} = q_{A \cap C(\mathcal{Q})}$, $v_I^{\mathcal{Q}} = q_I$ and $v_O^{\mathcal{Q}} \neq q_O$.

Since R indexes the set of all possible functions, there exists r such that:

(i) r maps $V_A = v_A$ to $V_B = v_B$ for all (v_A, v_B) such that $h(v_A) = v_B$ i.e. $r \in R_h$.

Algorithm 2 Procedure to efficiently construct LPs (21), (22)

Input: (i) causal graph G (ii) query $\mathcal{Q}_W = \mathbb{P}(V_O(V_I = q_I) = q_O | V_A = q_A, V_W = q_W)$ (iii) conditional probability distribution $p_{v_B.v_A} = \mathbb{P}(V_B = v_B | V_A = v_A)$, for all v_A, v_B.
Output: Pruned LPs (21) and (22)
$H \leftarrow \emptyset$
for $h : V_A \rightarrow V_B$ **do**
  if *h is valid (Theorem 5)* **then**
    $H \leftarrow H \cup \{h\}$
    Compute $\mathbf{1}\{R_h \subseteq R_{\mathcal{W}}\}$ using Theorem a
    Compute d_h^L using Theorems 6 and a
    Compute d_h^U using Theorems 7 and b
return *LPs* (21) *and* (22) *constructed using* $(H, \{\mathbf{1}\{R_h \subseteq R_{\mathcal{W}}\}\}_{h \in H}, d^L, d^U)$

(ii) r maps $V_{C(\mathcal{W})} = q_{C(\mathcal{W})}$ to $V_W = q_W$. This does not contradict i since there cannot exist $v^{\mathcal{W}} \in \{0,1\}^{|N|}$ such that $h(v_A^{\mathcal{W}}) = v_B^{\mathcal{W}}$, $v_{C(\mathcal{W})}^{\mathcal{W}} = q_{C(\mathcal{W})}$ and $v_W^{\mathcal{W}} \neq q_W$.

(iii) r maps $V_{A \cap C(\mathcal{Q})} = q_{A \cap C(\mathcal{Q})}$ and $V_I = q_I$ to $V_O = q_O$. This does not contradict i since there cannot exist $v^{\mathcal{Q}} \in \{0,1\}^{|N|}$ such that $h(v_A^{\mathcal{Q}}) = v_B^{\mathcal{Q}}$, $v_{A \cap C(\mathcal{Q})}^{\mathcal{Q}} = q_{A \cap C(\mathcal{Q})}$, $v_I^{\mathcal{Q}} = q_I$ and $v_O^{\mathcal{Q}} \neq q_O$, and it does not contradict ii since $R_{\mathcal{Q}} \cap R_{\mathcal{W}} \neq \emptyset$.

Hence, $R_h \cap (R_{\mathcal{Q}} \cap R_{\mathcal{W}}) \neq \emptyset$, a contradiction. ■

Thus, we have established that $\mathbf{1}\{R_h \subseteq R_{\mathcal{W}}\}$ can be efficiently computed using Theorem 11 (a), $d_h^L = \mathbf{1}\{R_h \subseteq R_{\mathcal{W}}\}\mathbf{1}\{R_h \subseteq R_{\mathcal{Q}}\}$ can be efficiently computed using Theorem 11 (a) and Theorem 6, and $d_h^U = 1 - \mathbf{1}\{R_h \cap (R_{\mathcal{W}} \cap R_{\mathcal{Q}}) = \emptyset\}$ can be efficiently computed using Theorem 7 and Theorem 11 (b) provided $R_{\mathcal{Q}} \cap R_{\mathcal{W}} \neq \emptyset$, i.e. the observation does not invalidate the query. Algorithm 2 describes a procedure that uses results established in Section 5.1 to efficiently construct pruned LPs (21) and (22) without formulating the original LPs or iterating over R.

6. Numerical Experiments

In this section, we report the results of our numerical experiments validating and extending the benefits from the new methods proposed in Sections 3 and 5.

6.1 Run time improvements

In this section, we numerically verify the computational savings from using the structural results in Sections 3 and 5 compared to the benchmark methods that iterate over the set R to compute the pruned LPs.

In Table 2 we report the results of the experiments for constructing the pruned LPs (5) and (7). Here t_H denotes the time (in seconds) taken by Algorithm 1 in Section 3.4, and t_R denotes the time (in seconds) taken by Algorithm 3, the benchmark that iterates over the set R. The results clearly show the significant runtime improvement provided by

Algorithm 3 Benchmark R-based method to construct LPs (5), (7)

Input: (i) Causal graph G (ii) Query $\mathcal{Q} = \mathbb{P}(V_O(V_I = q_I) = q_O | V_A = q_A)$ (iii) Conditional probability distribution $p_{v_B.v_A} = \mathbb{P}(V_B = v_B | V_A = v_A)$, for all v_A, v_B
Output: Pruned LPs (5) and (7)
$R_h \leftarrow \emptyset, \forall h : V_A \to V_B$
$c_h^L \leftarrow 1, c_h^U \leftarrow 0, \forall h : V_A \to V_B$
for $r \in R$ **do**
  Compute $F_B(V_A = v_A, r), \forall v_A \in \{0,1\}^{|A|}$
  Compute $F_O((V_A, V_I) = (q_A, q_I), r)$
  $R_{\bar{h}} \leftarrow R_{\bar{h}} \cup \{r\}$ for the hyperarc $\bar{h} : v_A \mapsto F_B(V_A = v_A, r)$
  if $c_{\bar{h}}^L = 1$ *and* $F_O((V_A, V_I) = (q_A, q_I), r) = 0$ **then**
    $c_{\bar{h}}^L \leftarrow 0$
  if $c_{\bar{h}}^U = 0$ *and* $F_O((V_A, V_I) = (q_A, q_I), r) = 1$ **then**
    $c_{\bar{h}}^U \leftarrow 1$
$H \leftarrow \{h : R_h \neq \emptyset\}$
return *LPs* (5) *and* (7) *constructed using* (H, c^L, c^U)

Graph	t_H (s)	t_R (s)
Ex A	2.0	> 1200
Ex B	3.9	72.6
Ex C	3.7	> 1200
Ex F	0.6	> 1200
Ex G	2.4	> 1200

Table 2: Runtime of Algorithm 1 in Section 3.4 to construct the pruned LPs (5) and (7) as compared to Algorithm 3, the benchmark that iterates over the set R for Examples A, B, C, F and G (details in Appendix B). Here t_H (resp. t_R) denotes the runtime in seconds of our proposed method (resp. iterating over R).

our method compared to the benchmark on the five examples. Note that the benchmark methods were terminated after 1200 seconds.

In Table 3 we report the runtimes for constructing the pruned LPs (21) and (22). Here t_H denotes the time (in seconds) taken by Algorithm 2 in Section 5.1 and t_R denotes the time (in seconds) taken by Algorithm 4 that iterates over the set R. The results clearly show the significant runtime improvement provided by our method compared to the benchmark on the five examples. Note that the benchmark methods were terminated after 1200 seconds.

6.2 Finite Data Setting

In this section, we show how to extend the pruning procedure to LPs where the conditional probabilities $p_{v_B.v_A}$ are estimated from finite amount of data, and therefore, are known only within a tolerance. Let $\bar{p}$ denote a sample estimate of p. Suppose the (unknown) true conditional probability $p_{v_B.v_A} \in [\bar{p}_{v_B.v_A} - \delta, \bar{p}_{v_B.v_A} + \delta]$ with high confidence. The bounds in

Algorithm 4 Benchmark R-based method to construct LPs (21) and (22)

Input: (i) causal graph G (ii) query $\mathcal{Q} = \mathbb{P}(V_O(V_I = q_I) = q_O | V_A = q_A, V_W = q_W)$ (iii) conditional probability distribution $p_{v_B.v_A} = \mathbb{P}(V_B = v_B | V_A = v_A)$, for all v_A, v_B

Output: Pruned LPs (21) and (22)

$H \leftarrow \emptyset$

$R_h \leftarrow \emptyset, \forall h : V_A \rightarrow V_B$

$c_h^L \leftarrow 1, c_h^U \leftarrow 0, \forall h : V_A \rightarrow V_B$

$d_h^L \leftarrow 1, d_h^U \leftarrow 0, \forall h : V_A \rightarrow V_B$

for $r \in R$ **do**

- Compute $F_B(V_A = v_A, r), \forall v_A \in \{0,1\}^{|A|}$
- Compute $F_O((V_A, V_I) = (q_A, q_I), r)$
- Compute $F_W(V_A = q_A, r)$
- $R_{\bar{h}} \leftarrow R_{\bar{h}} \cup \{r\}$ for the hyperarc $\bar{h} : v_A \mapsto F_B(V_A = v_A, r)$
- **if** $c_{\bar{h}}^L = 1$ *and* $F_O((V_A, V_I) = (q_A, q_I), r) = 0$ **then**
  - $c_{\bar{h}}^L \leftarrow 0$
- **if** $c_{\bar{h}}^U = 0$ *and* $F_O((V_A, V_I) = (q_A, q_I), r) = 1$ **then**
  - $c_{\bar{h}}^U \leftarrow 1$
- **if** $d_{\bar{h}}^L = 1$ *and* $F_W(V_A = q_A, r) = 0$ **then**
  - $d_{\bar{h}}^L \leftarrow 0$
- **if** $d_{\bar{h}}^U = 0$ *and* $F_W(V_A = q_A, r) = 1$ **then**
  - $d_{\bar{h}}^U \leftarrow 1$

$H \leftarrow \{h : R_h \neq \emptyset\}$

for $h \in H$ **do**

- $\mathbf{1}\{R_h \subseteq R_{\mathcal{W}}\} \leftarrow d_h^L$
- $d_h^L \leftarrow d_h^L \times c_h^L$
- $d_h^U \leftarrow d_h^U \times c_h^U$

return *LPs* (21) *and* (22) *constructed using* $(H, d^L, d^U, \{\mathbf{1}\{R_h \subseteq R_{\mathcal{W}}\}\}_{h \in H})$

Graph	t_H (s)	t_R (s)
Ex A	2.1	> 1200
Ex B	4.5	74.4
Ex C	4.3	> 1200
Ex F	0.7	> 1200
Ex G	3.1	> 1200

Table 3: Runtime of Algorithm 2 in Section 5.1 to construct the pruned LPs (21) and (22) as compared to Algorithm 4, the benchmark that iterates over the set R for Examples A, B, C, F and G (details in Appendix B). Here t_H (resp. t_R) denotes the runtime in seconds of our proposed method (resp. iterating over R).

this setting can be computed by solving the following LPs:

$$\begin{aligned} \alpha_L^F/\alpha_U^F = \quad & \min / \max_{q,p} \quad \textstyle\sum_{r:r\in R_{\mathcal{Q}}} q_r \\ & \text{s.t.} \quad \textstyle\sum_{r\in R_{v_B.v_A}} q_r \le \bar{p}_{v_B.v_A} + \delta, \quad \forall v_A, v_B \\ & \textstyle\sum_{r\in R_{v_B.v_A}} q_r \ge \bar{p}_{v_B.v_A} - \delta, \quad \forall v_A, v_B \\ & \textstyle\sum_{r\in R} q_r = 1, \\ & q \ge 0. \end{aligned} \tag{24}$$

Note that the structure of the objective function and the constraints in the q variables is the same as those in (3). Therefore, the results in Section 3 can be used to aggregate the q-variables in (24) and rewrite the LPs in terms of variables corresponding to hyperarcs.

$$\begin{aligned} \alpha_L^F/\alpha_U^F = \quad & \min / \max_{q,p} \quad \textstyle\sum_{h\in H} c_h^{L/U} q_h \\ & \text{s.t.} \quad \textstyle\sum_{h:h(v_A)=v_B} q_h \le \bar{p}_{v_B.v_A} + \delta, \quad \forall v_A, v_B \\ & \textstyle\sum_{h:h(v_A)=v_B} q_h \ge \bar{p}_{v_B.v_A} - \delta, \quad \forall v_A, v_B \\ & \textstyle\sum_{h\in H} q_h = 1 \\ & q \ge 0. \end{aligned} \tag{25}$$

In Figure 5 we plot the upper and lower bounds computed by solving (25) as a function of δ for 10 randomly generated instances of $\bar{p}$ in Example A. Note that as the uncertainty in p increases, the computed bounds become wider. Recall that for this example optimal bounds can only be computed after pruning the LP. Furthermore, standard results in LP duality (Bertsimas and Tsitsiklis, 1997) imply that one can utilize the optimal duals to identify $\bar{p}_{v_B.v_A}$ values that impact the bounds most, and concentrate additional measurement on these values. We would like to reiterate that the analysis presented here is only possible because we were able to prune the benchmark R-based LP for Example A.

We now extend LPs (21) and (22) to the finite data setting. Bounds in this setting can be computed by solving the following LPs:

$$\begin{aligned} \min_q / \max_q \quad & \frac{\sum_{r\in R_{\mathcal{Q}}\cap R_{\mathcal{W}}} q_r}{\sum_{r\in R_{\mathcal{W}}} q_r} \\ \text{s.t.} \quad & \textstyle\sum_{r\in R_{v_B.v_A}} q_r \le \bar{p}_{v_B.v_A} + \delta, \quad \forall (v_A, v_B), \\ & \textstyle\sum_{r\in R_{v_B.v_A}} q_r \ge \bar{p}_{v_B.v_A} - \delta, \quad \forall (v_A, v_B), \\ & \textstyle\sum_{r\in R} q_r = 1, \ q \ge 0. \end{aligned} \tag{26}$$

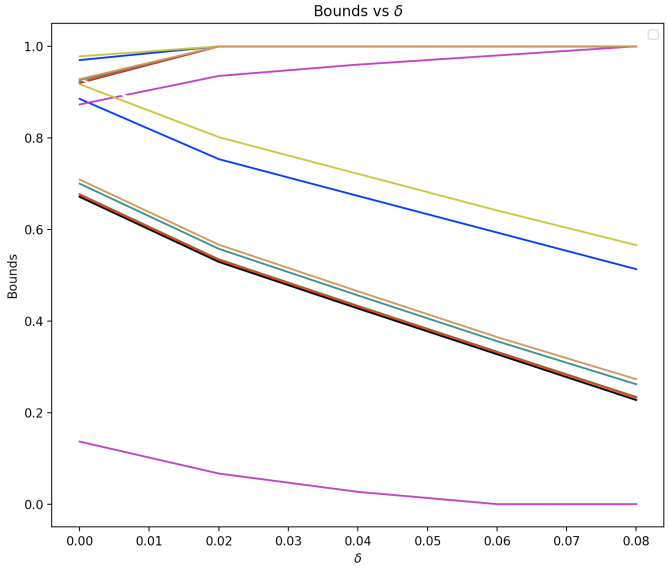


Figure 5: Lower and upper bounds vs δ for 10 instances of $\bar{p}$ in Example A. Lines of the same color denotes lower and upper bounds for the same instance.

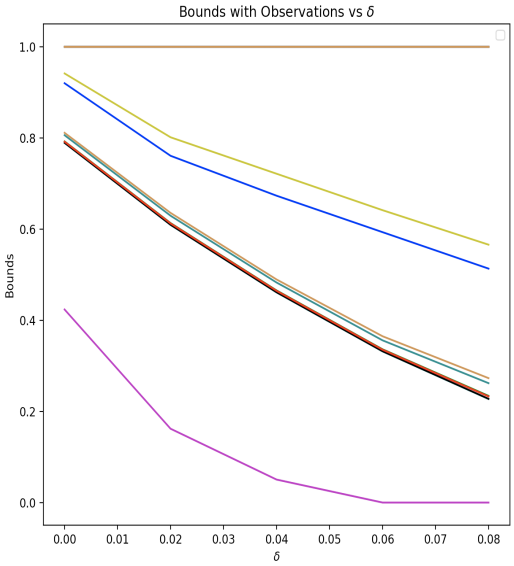
This fractional LP can be linearized as follows.

$$\begin{aligned} \min_{\alpha,q} / \max_{\alpha,q} \quad & \textstyle\sum_{r \in R_{\mathcal{Q}} \cap R_{\mathcal{W}}} q_r \\ \text{s.t.} \quad & \textstyle\sum_{r \in R_{v_B \cdot v_A}} q_r \leq (\bar{p}_{v_B \cdot v_A} + \delta)\alpha, \quad \forall (v_A, v_B), \\ & \textstyle\sum_{r \in R_{v_B \cdot v_A}} q_r \geq (\bar{p}_{v_B \cdot v_A} - \delta)\alpha, \quad \forall (v_A, v_B), \\ & \textstyle\sum_{r \in R} q_r = \alpha, \\ & \textstyle\sum_{r \in R_{\mathcal{W}}} q_r = 1, \\ & \alpha, q \geq 0. \end{aligned} \tag{27}$$

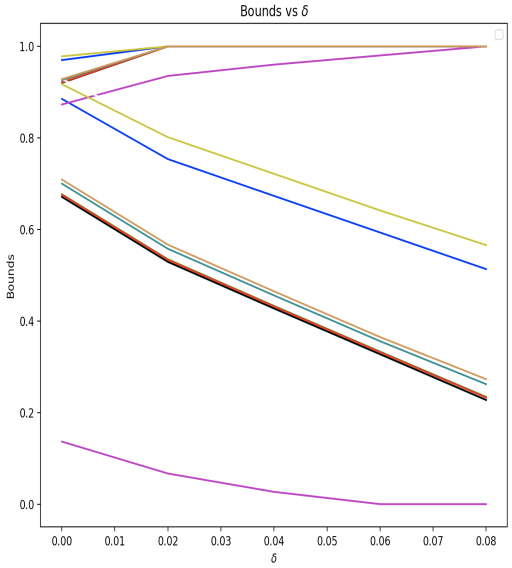
Note that this is a linear program in (α, q). The previous results allow us to aggregate the q-variables in (27) and rewrite the LPs in terms of variables corresponding to hyperarcs.

$$\begin{aligned} \min / \max_{q,p} \quad & \textstyle\sum_{h \in H} d_h^{L/U} q_h \\ \text{s.t.} \quad & \textstyle\sum_{h : h(v_A) = v_B} q_h \leq (\bar{p}_{v_B \cdot v_A} + \delta)\alpha, \quad \forall (v_A, v_B), \\ & \textstyle\sum_{h : h(v_A) = v_B} q_h \geq (\bar{p}_{v_B \cdot v_A} - \delta)\alpha, \quad \forall (v_A, v_B), \\ & \textstyle\sum_{h \in H} q_h = \alpha, \\ & \textstyle\sum_{h \in H : R_h \subseteq R_{\mathcal{W}}} q_h = 1, \\ & \alpha, p \geq 0. \end{aligned} \tag{28}$$

Recall that the lower and upper bounds computed by solving LPs (28) incorporate additional observations of some variables in V_B. In Figure 6a we plot these bounds as a function of δ for the same 10 instances of $\bar{p}$, but for Example A with additional observations. (details in Appendix B). We also replicate Figure 5 with bounds computed without these additional observations for comparison; note that our bounds have shifted significantly after the observation. As in Balke and Pearl (1994), we have numerically verified the distinction between causal inference for the entire population and the sub-population consisting of units consistent with the observation.



(a) Lower and upper bounds with additional observations vs δ for 10 instances of $\bar{p}$ in Example A. Lines of the same color denotes lower and upper bounds for the same instance.



(b) Lower and upper bounds without additional observations vs δ for 10 instances of $\bar{p}$ in Example A. Lines of the same color denotes lower and upper bounds for the same instance.

Figure 6: Lower and upper bounds with and without additional observations of some variables in V_B vs δ for 10 instances of $\bar{p}$ in Example A (details in Appendix B). Note that our bounds have shifted after the observation.

6.3 Greedy Heuristic

In this section, we propose a greedy heuristic (Algorithm 5) to approximately compute the bounds for problems *without* additional observations when even the pruned LPs are intractable, and the bounds cannot be computed in closed form since $A \not\subseteq C(\mathcal{Q})$. Note that the bounds computed by our heuristic are guaranteed to contain the optimal bounds, and therefore, the true query value. This heuristic is motivated by the duals of LPs (5) and (7) that are defined as follows:

$$\begin{aligned} \alpha_L = \quad & \max_{\lambda} \quad \textstyle\sum_{(v_A,v_B)\in\{0,1\}^{|A|}\times\{0,1\}^{|B|}} p_{v_B.v_A}\lambda_{v_B.v_A} \\ & \text{s.t.} \quad \textstyle\sum_{v_A\in\{0,1\}^{|A|}} \lambda_{h(v_A).v_A} \leq c_h^L, \qquad \forall h \in H. \end{aligned} \tag{29}$$

$$\begin{aligned} \alpha_U = \quad & \min_{\lambda} \quad \textstyle\sum_{(v_A,v_B)\in\{0,1\}^{|A|}\times\{0,1\}^{|B|}} p_{v_B.v_A}\lambda_{v_B.v_A} \\ & \text{s.t.} \quad \textstyle\sum_{v_A\in\{0,1\}^{|A|}} \lambda_{h(v_A).v_A} \geq c_h^U, \qquad \forall h \in H. \end{aligned} \tag{30}$$

We utilize the fact that in our numerical experiments, we observed that for both dual LPs, there was always an optimal solution that only took values in the set $\{-1, 0, 1\}$, and the fact that in the symbolic bounds introduced by Balke and Pearl (1994) (see, also Zhang and Bareinboim (2017); Pearl (2009); Sjölander et al. (2014); Sachs et al. (2022)) the probabilities in the input data were combined using coefficients taking values in $\{-1, 0, 1\}$. In fact, we expect the following conjecture to be true.

Graph	% instances with $\alpha_L^G = \alpha_L$	% instances with $\alpha_U^G = \alpha_U$	% instances with $\epsilon_L \leq 10\%$	% instances with $\epsilon_U \leq 10\%$
Ex A	100	100	100	100
Ex B	99	86	99	94
Ex C	100	84	100	86
Ex F	100	100	100	100
Ex G	100	100	100	100

Table 4: Quality of the bounds computed by the greedy algorithm over 100 instances of Examples A, B, C, F and G (details in Appendix B). Here α_L^G (resp. α_U^G) denotes the lower (resp. upper) bound computed by the greedy heuristic, α_L (resp. α_U) denotes the lower (resp. upper) bound computed by solving the LP, $\epsilon_L = (\alpha_L - \alpha_L^G)/\alpha_L$ (resp. $\epsilon_U = (\alpha_U^G - \alpha_U)/\alpha_U$) is a measure of suboptimality of the lower (resp. upper) bound computed by the greedy heuristic.

Algorithm 5 Greedy Heuristic

Let the permutation which sorts the conditional probabilities in descending order be $i_1, \ldots, i_{2^{|N|}}$.

Function `GreedyLowerBound()`:

```
Initialize λ = −1.
for j = 1, .., 2^|N| do
    while λ is feasible do
        λ_{i_j} = λ_{i_j} + 1
```

Function `GreedyUpperBound()`:

```
Initialize λ = 1.
for j = 1, .., 2^|N| do
    while λ is feasible do
        λ_{i_j} = λ_{i_j} − 1
```

Conjecture 13 (Dual Integrality) *The dual LPs in (29) and (30) have optimal solutions which only take values in* $\{-1, 0, 1\}$.

We tested Algorithm 5 on 100 instances of each of the examples in Appendix B for which bounds were not available in closed form. The results reported in Table 4 are for Examples A, B, C, F and G for which the LP can be solved. We see that bounds from the greedy heuristic matches the LP bounds in most instances for these problems. Recall that one can compute the optimal bounds for Examples A, F and G only after pruning the LP. In Table 4, $\epsilon_L = 1 - \frac{\alpha_L^G}{\alpha_L}$ and $\epsilon_U = \frac{\alpha_U^G}{\alpha_U} - 1$ denote the relative errors of the lower and upper bounds, respectively. We see that the lower bound is always within 10% of the true value, whereas the upper bound is within 10% for at least 86% of the cases. See Appendix for the empirical distribution function of errors. Furthermore, the greedy heuristic yields non-trivial

bounds for Examples D and E, where the corresponding pruned LP is too large to be solved to optimality.

7. Conclusion

In this work, we compute bounds for the expected value of some outcome variables V_O if we intervene on variables V_I, given the values of variables V_A are known, via linear programming. We show how to leverage structural properties of these LPs to significantly reduce their size. We also show how to construct these LPs efficiently. As a direct consequence of our results, bounds for causal queries can be computed for graphs of much larger size. We show that there are examples of causal inference problems for which bounds could be computed only after the pruning we introduce. Our structural results also allow us to characterize a set of causal inference problems for which the bounds can be computed in closed form. This class includes as a special case extensions of problems considered in the multiple confounded treatments literature (Wang and Blei, 2021; Ranganath and Perotte, 2019; Janzing and Schölkopf, 2018; D'Amour, 2019; Tran and Blei, 2017). We show that bounds for queries containing additional observations about the unit can be computed by solving fractional LPs (Bitran and Novaes, 1973). These fractional LPs are special because the denominator is restricted to be non-negative. This allows us to homogenize the problem into a LP with one additional constraint, and extend the structural results obtained for queries without additional observations. We also show the significant runtime improvement provided by our methods compared to benchmarks in numerical experiments and extend the results to the finite data setting. Finally, for causal inference without additional observations, we propose a very efficient greedy heuristic that produces very high quality bounds, and scales to problems that are several orders of magnitude larger than those for which the pruned LPs can be solved.

Appendix A. Basic results

Lemma 14 (Partition of R) *$R = \cup_{h \in H} R_h$ is a partition of R.*

Proof Since $R_h \subseteq R$ for all $h \in H$, it follows that $\cup_{h \in H} R_h \subseteq R$. Next, we show that $R \subseteq \cup_h R_h$. Fix $\bar{r} \in R$. Then $\bar{r} \in R_h$ for some hyperarc h such that $h(v_A) = F_B(v_A, \bar{r})$ for all $v_A \in \{0,1\}^{|A|}$. Thus, $R \subseteq \cup_h R_h$.

Next, suppose there exist $h_1 \neq h_2$ such that $R_{h_1} \cap R_{h_2} \neq \emptyset$. Then for all $r \in R_{h_1} \cap R_{h_2}$, and all $v_A \in \{0,1\}^{|A|}$, we have $h_i(v_A) = F_B(v_A, r)$, $i = 1, 2$. Thus, it follows that $h_1(v_A) = h_2(v_A)$ for all $v_A \in \{0,1\}^{|A|}$. A contradiction. ■

Lemma 15 (Critical Intervention Variables) *Let $G^{do(V_I = q_I)}$ denote the mutilated graph after intervention $do(V_I = q_I)$, i.e. variables V_I no longer have any incoming arcs, and let $V_{C(\mathcal{Q})}$ denote the set of variables in $V_A \cup V_B$ that have a path to some variable in V_O in $G^{do(V_I = q_I)}$. Then*

$$\mathbb{P}(V_O(V_I = q_I) = q_O | V_A = q_A) = \mathbb{P}(V_O(V_{I \cap C(\mathcal{Q})} = q_{I \cap C(\mathcal{Q})}) = q_O | V_A = q_A)$$

Proof By definition, for every $V \in V_{I \setminus C(\mathcal{Q})}$, there is no directed path from V to a variable in V_O in $G^{do(V_I = q_I)}$. It thus follows that they do not influence the value of V_O in the intervention. It thus follows that $\mathbb{P}(V_O(V_I = q_I) = q_O | V_A = q_A) = \mathbb{P}(V_O(V_{I \cap C(\mathcal{Q})} = q_{I \cap C(\mathcal{Q})}) = q_O | V_A = q_A)$. ■

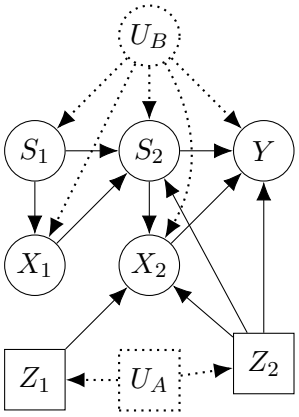
Appendix B. Examples of Causal Inference Problems

In this section, we report the causal graph structure and the data generation process for the 5 examples in Table 1.

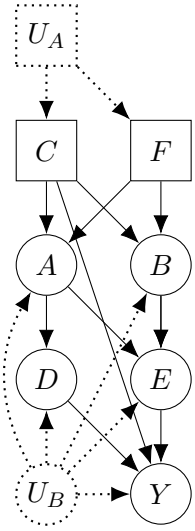
Example A

The causal graph for this example is display in Figure 7a. The query is: $P(Y(X_2 = 1) = 1 | Z_1 = 1, Z_2 = 1)$, and data generating process used to generate the input data is given by

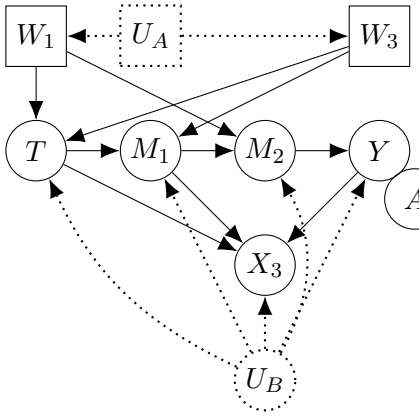
$$\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
Z_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
Z_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
S_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B)) \\
X_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + S_1)) \\
S_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(S_1 + U_B + X_1 + Z_2)) \\
X_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(S_2 + U_B + Z_1 + Z_2)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + S_2 + X_2 + Z_2))
\end{aligned}$$



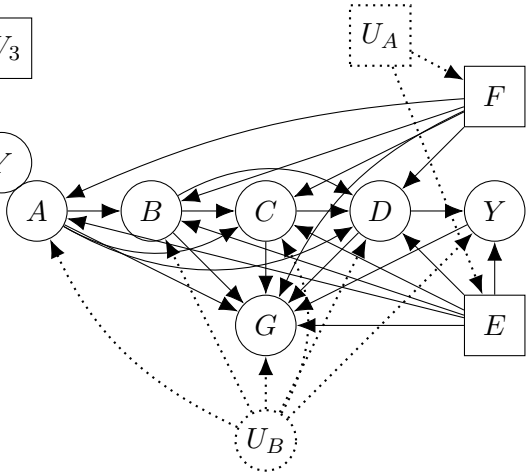
(a) Example A



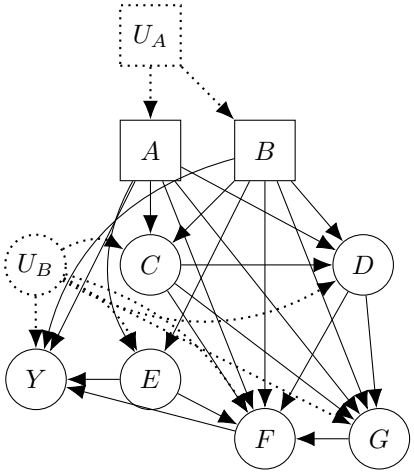
(b) Example B



(c) Example C

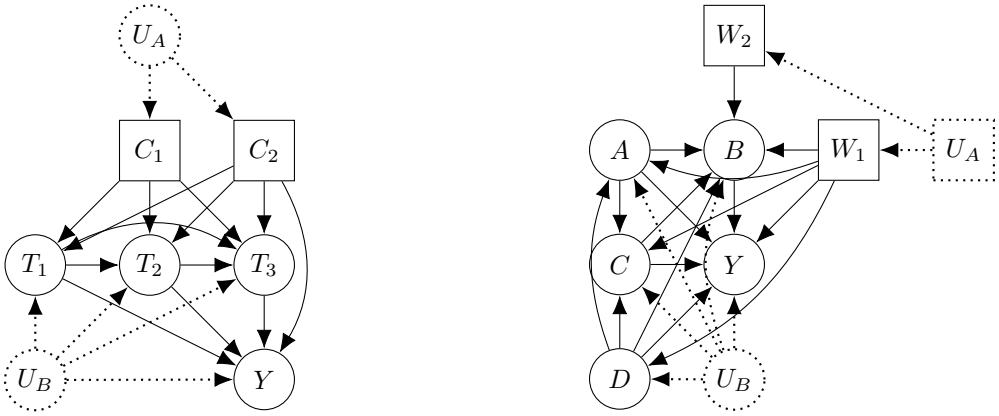


(d) Example D



(e) Example E

Figure 7: Examples of Causal Inference Problems



(a) Example F (b) Example G

Figure 8: Additional Causal Inference Problems

After sampling U_A, U_B, Z_1, Z_2 we compute
$\mathbb{P}(Y, X_2, S_2, X_1, S_1 | Z_2, Z_1) = \mathbb{P}(Y|U_B, S_2, X_2, Z_2)\mathbb{P}(X_2|S_2, Z_1, Z_2, U_B)\mathbb{P}(S_2|Z_2, U_B, S_1, X_1)\mathbb{P}(X_1|S_1, U_B)$
$\mathbb{P}(S_1|U_B)$
that gives input distribution. In Section 6.3, we use the observation $\{Z_1 = 1, Z_2 = 1, Y = 1\}$.

Example B

The causal graph for this example is in Figure 7b and the query is: $P(Y(A = 1, B = 1) = 1|C = 1, F = 1)$. The data generating process used to generate the input information is as follows:

$$
\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
C &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
F &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
A &\sim \text{Bernoulli}(\text{logit}^{-1}(C + F + U_B)) \\
B &\sim \text{Bernoulli}(\text{logit}^{-1}(C + F + U_B)) \\
D &\sim \text{Bernoulli}(\text{logit}^{-1}(A + U_B)) \\
E &\sim \text{Bernoulli}(\text{logit}^{-1}(A + B + U_B)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + D + C + E))
\end{aligned}
$$

After sampling U_A, U_B, C, F we compute

$$\mathbb{P}(A, B, D, E, Y|C, F) = \mathbb{P}(Y|U_B, D, C, E)\mathbb{P}(E|A, B, U_B)\mathbb{P}(D|A, U_B)\mathbb{P}(B|C, F, U_B)\mathbb{P}(A|C, F, U_B).$$

that gives the input distribution. In Section 6.3, we use the observation $\{C = 1, F = 1, Y = 1\}$.

Example C

The causal graph for this example is in Figure 7c and the query is: $P(Y(M_1 = 1) = 1|W_1 = 1, W_3 = 1)$. The data generating process used to generate the input information is as follows:

$$\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
W_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
W_3 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
T &\sim \text{Bernoulli}(\text{logit}^{-1}(W_1 + W_3 + U_B)) \\
M_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(T + W_3 + U_B)) \\
M_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(M_1 + W_1 + U_B)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(M_2 + U_B)) \\
X_3 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + Y + M_1 + T))
\end{aligned}$$

After sampling U_A, U_B, W_1, W_3 we compute

$\mathbb{P}(T, M_1, M_2, X_3, Y|W_1, W_3) = \mathbb{P}(X_3|U_B, Y, M_1, T)\mathbb{P}(Y|M_2, U_B)\mathbb{P}(M_2|M_1, W_1, U_B)$ $\mathbb{P}(M_1|T, W_3, U_B)\mathbb{P}(T|W_1, W_3, U_B)$

that gives the input distribution. In Section 6.3, we use the observation $\{W_1 = 1, W_3 = 1, Y = 1\}$.

Example D

The causal graph for this example is in Figure 7d and the query is: $P(Y(D = 1) = 1|E = 1, F = 1)$. The data generating process used to generate the input information is as follows:

$$\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
F &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
E &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
A &\sim \text{Bernoulli}(\text{logit}^{-1}(F + E + U_B)) \\
B &\sim \text{Bernoulli}(\text{logit}^{-1}(E + F + A + U_B)) \\
C &\sim \text{Bernoulli}(\text{logit}^{-1}(B + E + F + A + U_B)) \\
D &\sim \text{Bernoulli}(\text{logit}^{-1}(B + E + F + A + C + U_B)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(E + D + U_B)) \\
G &\sim \text{Bernoulli}(\text{logit}^{-1}(A + B + C + D + Y + E + F + U_B))
\end{aligned}$$

After sampling U_A, U_B, E, F we compute

$\mathbb{P}(G, Y, D, C, B, A|E, F) = \mathbb{P}(G|U_B, A, B, C, D, Y, E, F)\mathbb{P}(Y|E, D, U_B)\mathbb{P}(D|B, E, F, A, C, U_B)$ $\mathbb{P}(C|B, E, F, A, U_B)\mathbb{P}(B|E, F, A, U_B)\mathbb{P}(A|E, F, U_B)$

that gives the input distribution. In Section 6.3, we use the observation $\{E = 1, F = 1, Y = 1\}$.

Example E

The causal graph for this example is in Figure 8 and the query is: $P(Y(C = 1, F = 1) = 1|A = 1, B = 1)$. The data generating process used to generate the input information is as follows:

$$\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
A &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
B &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
C &\sim \text{Bernoulli}(\text{logit}^{-1}(A + B + U_B)) \\
D &\sim \text{Bernoulli}(\text{logit}^{-1}(A + C + B + U_B)) \\
E &\sim \text{Bernoulli}(\text{logit}^{-1}(A + B + U_B)) \\
F &\sim \text{Bernoulli}(\text{logit}^{-1}(A + C + B + D + E + G + U_B)) \\
G &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + A + B + C + D)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + A + E + B + F))
\end{aligned}$$

After sampling U_A, U_B, A, B we compute
$\mathbb{P}(C, D, E, F, G, Y|A, B) = \mathbb{P}(C|A, B, U_B)\mathbb{P}(D|A, C, B, U_B)\mathbb{P}(E|A, B, U_B)\mathbb{P}(F|A, C, B, D, E, G, U_B)$
$\mathbb{P}(G|A, B, C, D, U_B)\mathbb{P}(Y|U_B, A, E, B, F)$
that gives the input distribution. In Section 6.3, we use the observation $\{A = 1, B = 1, Y = 1\}$.

Example F

The causal graph for this example is in Figure 8a and the query is: $P(Y(T_1 = 1, T_2 = 1, T_3 = 1) = 1|C_1 = 1, C_2 = 1)$. The data generating process used to generate the input information is as follows:

$$\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
C_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
C_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
T_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(C_1 + C_2 + U_B)) \\
T_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(C_1 + C_2 + U_B)) \\
T_3 &\sim \text{Bernoulli}(\text{logit}^{-1}(C_1 + C_2 + U_B)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + T_1 + T_2 + T_3 + C_2))
\end{aligned}$$

After sampling U_A, U_B, C_1, C_2 we compute
$\mathbb{P}(T_1, T_2, T_3|C_1, C_2) = \mathbb{P}(T_1|C_1, C_2, U_B)\mathbb{P}(T_2|C_1, C_2, U_B)\mathbb{P}(T_3|C_1, C_2, U_B)\mathbb{P}(Y|T_1, T_2, T_3, , C_2, U_B)$
that gives the input distribution. In Section 6.3, we use the observation $\{C_1 = 1, C_2 = 1, Y = 1\}$.

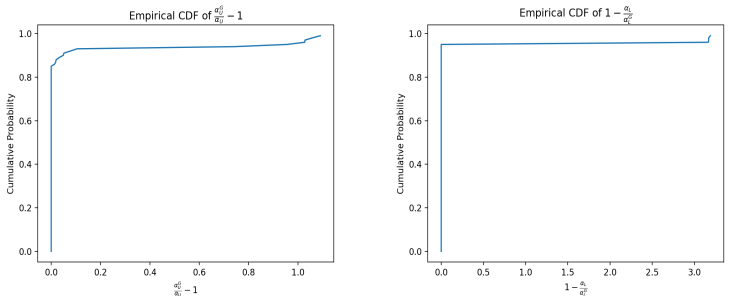
Example G

The causal graph for this example is in Figure 8b and the query is: $P(Y(B = 1) = 1|W_1 = 1, W_2 = 1)$. The data generating process used to generate the input information is as follows:

$$
\begin{aligned}
U_A &\sim N(0,1) \\
U_B &\sim N(0,1) \\
W_1 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
W_2 &\sim \text{Bernoulli}(\text{logit}^{-1}(U_A)) \\
D &\sim \text{Bernoulli}(\text{logit}^{-1}(W_1 + U_B)) \\
A &\sim \text{Bernoulli}(\text{logit}^{-1}(W_1 + D + U_B)) \\
C &\sim \text{Bernoulli}(\text{logit}^{-1}(A + D + W_1 + U_B)) \\
B &\sim \text{Bernoulli}(\text{logit}^{-1}(D + C + A + W_1 + W_2 + U_B)) \\
Y &\sim \text{Bernoulli}(\text{logit}^{-1}(U_B + D + C + A + B + W_1))
\end{aligned}
$$

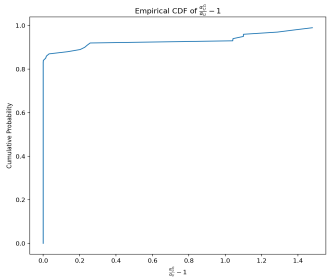
After sampling U_A, U_B, W_1, W_2 we compute
$\mathbb{P}(D, A, C, B, Y|W_1, W_2) = \mathbb{P}(D|W_1, U_B)\mathbb{P}(A|W_1, D, U_B)\mathbb{P}(C|A, D, W_1, U_B)\mathbb{P}(B|D, C, A, W_1, W_2, U_B)$
$\mathbb{P}(Y|D, C, A, B, W_1, U_B)$
that gives the input distribution. In Section 6.3, we use the observation $\{W_1 = 1, W_2 = 1, Y = 1\}$.

Appendix C. Empirical CDF for Error of Greedy Heuristic



(a) Empirical CDF of the Relative Error of α_U for Example B

(b) Empirical CDF for Relative Error of α_L for Example B



(c) Empirical CDF of the Relative Error of α_U for Example C

Figure 9: Empirical Distribution Functions of Errors for Examples

References

Holly Andersen. When to expect violations of causal faithfulness and why it matters. *Philosophy of Science*, 80(5):672–683, 2013.

A. Balke and J. Pearl. Counterfactual probabilities: Computational methods, bounds and applications. In *Uncertainty Proceedings 1994*, pages 46–54. Elsevier, 1994.

Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1st edition, 1997. ISBN 1886529191.

G. R. Bitran and A. G. Novaes. Linear programming with a fractional objective function. *Operations Research*, 21(1):22–29, 1973.

A. Charnes and W. W. Cooper. Programming with linear fractional functionals. *Naval Research logistics quarterly*, 9(3-4):181–186, 1962.

A. D'Amour. On multi-cause approaches to causal inference with unobserved counfounding: Two cautionary failure cases and a promising alternative. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 3478–3486. PMLR, 16–18 Apr 2019. URL `https://proceedings.mlr.press/v89/d-amour19a.html`.

G. Duarte, N. Finkelstein, D. Knox, J. Mummolo, and I. Shpitser. An automated approach to causal inference in discrete settings. *arXiv preprint arXiv:2109.13471*, 2021.

R. J. Evans. Graphical methods for inequality constraints in marginalized DAGs. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2012. doi: 10.1109/MLSP.2012.6349796.

N. Finkelstein and I. Shpitser. Deriving bounds and inequality constraints using logical relations among counterfactuals. In *Conference on Uncertainty in Artificial Intelligence*, pages 1348–1357. PMLR, 2020.

N. Finkelstein, R. Adams, S. Saria, and I. Shpitser. Partial identifiability in discrete data with measurement error. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1798–1808. PMLR, 27–30 Jul 2021. URL `https://proceedings.mlr.press/v161/finkelstein21b.html`.

D. Geiger and C. Meek. Quantifier elimination for statistical problems, 2013. URL `https://arxiv.org/abs/1301.6698`.

K. Imai and Z. Jiang. Discussion of "the blessings of multiple causes" by wang and blei, 2019.

G. W. Imbens and D. B. Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

D. Janzing and B. Schölkopf. Detecting confounding in multivariate linear models via spectral analysis. *Journal of Causal Inference*, 6(1):20170013, 2018. doi: doi:10.1515/jci-2017-0013. URL https://doi.org/10.1515/jci-2017-0013.

N. Kilbertus, M. J. Kusner, and R. Silva. A class of algorithms for general instrumental variable models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20108–20119. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e8b1cbd05f6e6a358a81dee52493dd06-Paper.pdf.

E. L. Ogburn, I. Shpitser, and E. J. Tchetgen. Comment on "blessings of multiple causes". *Journal of the American Statistical Association*, 114(528):1611–1615, 2019. doi: 10.1080/01621459.2019.1689139. URL https://doi.org/10.1080/01621459.2019.1689139.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

D. Poderini, R. Chaves, I. Agresti, G. Carvacho, and F. Sciarrino. Exclusivity graph approach to instrumental inequalities. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 1274–1283. PMLR, 22–25 Jul 2020. URL https://proceedings.mlr.press/v115/poderini20a.html.

Rajesh Ranganath and Adler Perotte. Multiple causal inference with latent confounding, 2019.

A. Richardson, M. G. Hudgens, P.B. Gilbert, and J.P. Fine. Nonparametric bounds and sensitivity analysis of treatment effects. *Stat Sci*, 29(4):596–618, 2014. doi: 10.1214/14-STS499.

M. C. Sachs, E. E. Gabriel, and A. Sjolander. Symbolic computation of tight causal bounds. *Biometrika*, 103(1):1–19, 2020.

Michael C Sachs, Gustav Jonzon, Arvid Sjölander, and Erin E Gabriel. A general method for deriving tight symbolic bounds on causal effects. *Journal of Computational and Graphical Statistics*, pages 1–10, 2022.

Madhumitha Shridharan and Garud Iyengar. Scalable computation of causal bounds. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20125–20140. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/shridharan22a.html.

A. Sjölander, W. Lee, H. Källberg, and Y. Pawitan. Bounds on causal interactions for binary outcomes. *Biometrics*, 70(3):500–505, 2014. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/24538083.

D. Tran and D. M. Blei. Implicit causal models for genome-wide association studies, 2017.

Y. Wang and D. M. Blei. The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528):1574–1596, 2019a.

Y. Wang and D. M. Blei. A proxy variable view of shared confounding. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10697–10707. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wang21c.html.

Yixin Wang and David M. Blei. The blessings of multiple causes: Rejoinder. *Journal of the American Statistical Association*, 114(528):1616–1619, 2019b. doi: 10.1080/01621459.2019.1690841. URL https://doi.org/10.1080/01621459.2019.1690841.

J. Zhang and E. Bareinboim. Transfer learning in multi-armed bandits: A causal approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1340–1346, 2017. doi: 10.24963/ijcai.2017/186. URL https://doi.org/10.24963/ijcai.2017/186.

J. Zhang and E. Bareinboim. Bounding causal effects on continuous outcome. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):12207–12215, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17449.

Junzhe Zhang, Jin Tian, and Elias Bareinboim. Partial counterfactual identification from observational and experimental data. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26548–26558. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zhang22ab.html.