An Algorithm and Complexity Results for Causal Unit Selection

Haiying Huang HHAIYING@CS.UCLA.EDU
UCLA Computer Science Department

Adnan Darwiche DARWICHE@CS.UCLA.EDU
UCLA Computer Science Department

Editors: Mihaela van der Schaar, Dominik Janzing and Cheng Zhang

Abstract

The unit selection problem aims to identify objects, called units, that are most likely to exhibit a desired mode of behavior when subjected to stimuli (e.g., customers who are about to churn but would change their mind if encouraged). Unit selection with counterfactual objective functions was introduced relatively recently with existing work focusing on bounding a specific class of objective functions, called the benefit functions, based on observational and interventional data—assuming a fully specified model is not available to evaluate these functions. We complement this line of work by proposing the first exact algorithm for finding optimal units given a broad class of causal objective functions and a fully specified structural causal model (SCM). We show that unit selection under this class of objective functions is NP^{PP}-complete but is NP-complete when unit variables correspond to all exogenous variables in the SCM. We also provide treewidth-based complexity bounds on our proposed algorithm while relating it to a well-known algorithm for Maximum a Posteriori (MAP) inference.

Keywords: unit selection, structural causal models, counterfactual reasoning

1. Introduction

A theory of causality has emerged over the last few decades based on two parallel hierarchies, an *information hierarchy* and a *reasoning hierarchy,* often called the *causal hierarchy* (Pearl and Mackenzie, 2018; Bareinboim et al., 2021). On the reasoning side, this theory has crystalized three levels of reasoning with increased sophistication and proximity to human reasoning: associational, interventional and counterfactual, which are exemplified by the following canonical probabilities. *Associational* $Pr(y|x)$: probability of y given that x was observed (e.g., probability that a patient has a flu given they have a fever). *Interventional* $Pr(y_x)$: probability of y given that x was established by an intervention, which is different from $Pr(y|x)$ (e.g., seeing the barometer fall tells us about the weather but moving the barometer needle won't bring rain). *Counterfactual* $Pr(y_x|y', x')$: probability of y if we were to establish x given that neither x nor y are true (e.g., probability that a patient who did not take a vaccine and died would have recovered had they been vaccinated). On the information side, these forms of reasoning require different levels of knowledge, encoded as associational, causal and functional (mechanistic) models, with each class of models containing more information than the preceding one. In the framework of probabilistic graphical models (Koller and Friedman, 2009), such knowledge is encoded by Bayesian networks (Pearl, 1988; Darwiche, 2009), causal Bayesian networks (Pearl, 2000; Peters et al., 2017; Spirtes et al., 2000) and functional Bayesian networks (Balke and Pearl, 1995) also known as *structural causal models* (SCMs).

One utility of this theory has been recently crystallized through the *unit selection problem* introduced by Li and Pearl (2019) who motivated it using the problem of selecting customers to target

by an encouragement offer for renewing a subscription. Let c denote the characteristics of a customer, x denote encouragement and y denote renewal. One can use counterfactuals to describe the different types of customers. A responder $(y_x, y'_{x'})$ would renew a subscription if encouraged but would not renew otherwise. An always-taker $(y_x, y_{x'})$ would always renew regardless of encouragement. An always-denier $(y'_x, y'_{x'})$ would always not renew regardless of encouragement. A contrarian $(y_{x'}, y'_x)$ would not renew if encouraged but would renew otherwise. One can then identify customers of interest by optimizing an expression, called a *benefit function* in (Li and Pearl, 2019), that includes counterfactual probabilities. In this example, the benefit function has the form $\beta Pr(\text{responder}|c) + \gamma Pr(\text{always-taker}|c) + \theta Pr(\text{always-denier}|c) + \delta Pr(\text{contrarian}|c)$ where $\beta, \gamma, \theta, \delta$ are corresponding benefits. In other words, one can use this expression to score customers with characteristics c so the most promising ones can be selected for an encouragement offer. When the above benefit function is contrasted with classical loss functions (for example, ones used to train neural networks), one sees a fundamental role for counterfactual reasoning as it gives us an ability to distinguish between objects (e.g., people, situations) depending on how they respond to a stimulus. This distinction sets apart counterfactual reasoning (third level of the causal hierarchy) from the more common, but less refined, associational reasoning (first level). It also sets it apart from interventional reasoning (second level) which is also not sufficient to make such distinctions.

Existing work on unit selection has focused on a very practical setting in which only the structure of an SCM is available together with some observational and experimental data (Li and Pearl, 2019, 2022a,b,c; Li et al., 2022b). Such data is usually not sufficient to obtain a fully specified SCM so one cannot obtain point values of the benefit function. Recent work has therefore focused on bounding probabilities of causation while tightening these bounds as much as possible (Dawid et al., 2017; Mueller et al., 2021), but with less attention dedicated to optimizing benefits based on these bounds; see (Li et al., 2022b,a) for a notable exception. In this paper, we complement this line of work by studying the unit selection problem from a different and computational direction. We are particularly interested in applying unit selection to structured units (e.g., decisions, policies, people, situations, regions, activities) that correspond to instantiations of multiple variables (called unit variables). We assume a fully specified SCM so we can obtain point values for any *causal objective function* as discussed in Section 2. By a causal objective function we mean any expression involving quantities from any level of the causal hierarchy (observational, interventional and counterfactual). This allows us to seek units that satisfy a broad class of conditions. Examples include: Which combination of activities are most effective to address a particular humanitarian need (human suffering, disease, hunger, privation)? Which regions should be focused on to reduce population movements among refugees? What incentive policy would keep customers engaged for the longest time? We then consider a particular but broad class of causal objective functions in Section 3 and formally define the computational problem of finding units that optimize these functions. We dedicate Section 4 to studying the complexity of unit selection in this setting where we show it has the same complexity as the classical *Maximum a Posteriori (MAP)* problem. We then provide an exact algorithm for solving the unit optimization problem in Section 5 by reducing it to a new problem that we call *Reverse-MAP*. We further characterize the complexity of our proposed algorithm using the notion of treewidth and provide some analysis on how its complexity can change depending on the specific objective function we use. We finally close with some concluding remarks in Section 6. Some proofs of our results are included in the main paper, the remaining ones can be found in the appendix.

2. Counterfactual Queries on Structural Causal Models

We review *structural causal models* (SCMs) in this section since the unit selection problem is defined on these models; see (Galles and Pearl, 1998; Halpern, 2000) for a comprehensive exposition. We use uppercase letters (e.g., X) to denote variables and lowercase letters (e.g., x) to denote their states. We use bold uppercase letters (e.g., $\mathbf{X}$) to denote sets of variables and bold lowercase letters (e.g., $\mathbf{x}$) to denote their instantiations. The states of a binary variable X are denoted x and x'. We also write $x \in \mathbf{x}$ to mean that variable X has state x in instantiation $\mathbf{x}$ of variables $\mathbf{X}$.

An SCM has three components. First, a directed acyclic graph with its nodes representing variables. Root nodes are called *exogenous* and internal nodes are called *endogenous*. Second, a probability distribution $\theta(U)$ for each exogenous variable U in the model. Third, for each endogenous variable V with parents $\mathbf{P}$, the SCM has an equation, called a *structural equation*, which specifies a state for V for each instantiation $\mathbf{p}$ of its parents $\mathbf{P}$. Let $\mathbf{U}/\mathbf{V}$ be the exogenous/endogenous variables in an SCM. The distribution $Pr(\mathbf{U}, \mathbf{V})$ specified by the SCM is as follows: $Pr(\mathbf{u}, \mathbf{v}) = \prod_{u \in \mathbf{u}} \theta(u)$ if $\mathbf{V} = \mathbf{v}$ is implied by $\mathbf{U} = \mathbf{u}$ and the structural equations; otherwise, $Pr(\mathbf{u}, \mathbf{v}) = 0$.

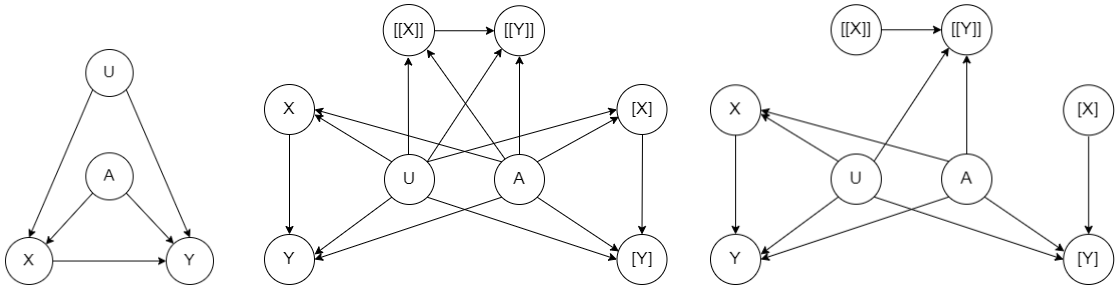
SCMs are a special type of Bayesian networks (Pearl, 1989; Darwiche, 2009) which require a conditional probability table (CPT) for each node in the network. In particular, for node V with parents $\mathbf{P}$, the CPT specifies the conditional distributions $Pr(V|\mathbf{P})$. A structural equation can be encoded as a CPT which satisfies $Pr(v|\mathbf{p}) \in \{0, 1\}$ for all v and $\mathbf{p}$. Such a CPT is said to be *functional* and this is why SCMs are sometimes called *functional Bayesian networks*.

A Bayesian network can only be used to compute *observational* probabilities such as $Pr(y|x)$ which is the probability of $Y = y$ given that we *observed* $X = x$. An SCM can also be used to compute *interventional probabilities* such as $Pr(y_x)$ which is the probability of $Y = y$ after *setting* $X = x$. An SCM can further be used to compute *counterfactual probabilities* such as $Pr(y_x, y'_{x'}|e)$ which is the probability of ($Y = y$ after setting $X = x$ and $Y = y'$ after setting $X = x'$) in a situation where we observe $E = e$.[1] We are particularly interested in this form of counterfactual probabilities as they will be used as ingredients in our objective functions. We next show how to compute such a counterfactual probability on an SCM by computing an observational probability on an auxiliary model. This will be essential for the constructions used later in the paper.

Consider the counterfactual probability $Pr(y_x, y'_{x'}|x, y)$ on the SCM in Figure 1(*a*). This query has three conflicting components: y_x, $y'_{x'}$ and (x, y). The first two involve conflicting actions (x and x'). Moreover, the actions and outcomes in the first two components conflict with the observation in the third component (x, y). This is why computing counterfactual probabilities usually requires an auxiliary model that incorporates multiple worlds (real and imaginary) that all share the same causal mechanisms (exogenous variables). For the counterfactual queries we are interested in, an auxiliary model with three worlds will suffice as we discuss next.

Given an SCM G, its *triplet model* is another SCM constructed by having three copies G^1, G^2 and G^3 of G and then joining them so they share their exogenous variables; see Figure 1(*b*). If X is a variable in G^1, we will use $[X]$ to denote its copy in G^2 and $[[X]]$ to denote its copy in G^3. A triplet model is a special case of *parallel worlds models* (Avin et al., 2005) which also include *twin models* (Balke and Pearl, 1994).[2] We can now compute the counterfactual probability $Pr(y_x, y'_{x'}|x, y)$ on SCM G by operating on the triplet model as follows. First, we mutilate copies

1. The class of *causal Bayesian networks* sits between Bayesian networks and functional Bayesian networks as it can be used to compute observational and interventional probabilities but not counterfactual ones (Pearl et al., 2000).
2. Twin models are sufficient to evaluate counterfactual probabilities like $Pr(y'_{x'}|x, y)$ and $Pr(y'_x, y_{x'})$ but not ones like $Pr(y_x, y'_{x'}|e)$ which we are interested in; see also (Tian and Pearl, 2000; Pearl et al., 2000).



(*a*) SCM (*b*) triplet model (*c*) mutilated triplet model

Figure 1: Reducing the counterfactual probability $Pr(y_x, y'_{x'}|x, y)$ on the model in (a) to an observational probability $Pr([y], [[y']] \mid [x], [[x']], x, y)$ on the model in (c).

G^2 and G^3 in the triplet model by removing the edges pointing into variables $[X]$ and $[[X]]$ and setting $[X] = x$ and $[[X]] = x'$ (since we are intervening on these variables). The result is a *mutilated triplet model* shown in Figure 1(*c*). We can then evaluate $Pr(y_x, y'_{x'}|x, y)$ on the SCM G by computing the observational probability $Pr([y], [[y']] \mid [x], [[x']], x, y)$ on the mutilated triplet model. Intuitively, the triplet model can be viewed as capturing three worlds G^1, G^2 and G^3. World G^1 captures the observation x, y; world G^2 captures the intervention $X = x$, and world G^3 captures the intervention $X = x'$. This above treatment can be directly generalized to counterfactual queries of the form $Pr(\mathbf{y_x}, \mathbf{w_v}|\mathbf{e})$ where $\mathbf{E}, \mathbf{X}, \mathbf{Y}, \mathbf{V}, \mathbf{W}$ are sets of variables. It is precisely this class of counterfactual queries that we shall use in the rest of the paper, starting with the next section.

3. Causal Objective Functions and Unit Selection

A causal objective function can be any expression that involves observational, interventional or counterfactual probabilities where the goal of unit selection is to find objects (units) that optimize this function. However, inspired by (Li and Pearl, 2019), our treatment will be based on a specific class of causal objective functions which is a linear combination of counterfactual probabilities of the form $Pr(\mathbf{y}^i{}_{\mathbf{x}^i}, \mathbf{w}^i{}_{\mathbf{v}^i}|\mathbf{e}^i, \mathbf{u})$ where $i = 1, \ldots, n$. We call $\mathbf{U}$ the *unit variables* since our goal is to find instantiations $\mathbf{u}$ of these variables (i.e., units) that optimize the objective function.[3] Variables $\mathbf{X}^i\mathbf{V}^i$ represent treatments, variables $\mathbf{Y}^i\mathbf{W}^i$ represent outcomes, $(\mathbf{X}^i \cup \mathbf{V}^i) \cap (\mathbf{Y}^i \cup \mathbf{W}^i) = \emptyset$, and variables $\mathbf{E}^i$ represent evidence. Unit variables are shared by all components of the objective function but each component can have its own treatment, outcome and evidence variables.

We will further assume that unit variables $\mathbf{U}$ are exogenous in the SCM (i.e., root variables) while treatment, outcome and evidence variables are endogenous. However, not all exogenous variables need to be unit variables. This is consistent with the assumption in (Li and Pearl, 2022a) that unit variables (also called *characteristics*) cannot be descendants of treatment or outcome variables.

3. An anonymous reviewer pointed out that the term "unit" is often used to designate the unit of analysis; that is, the entity that is characterized by random variables. For example, the unit in many medical studies is the the patient, the unit in many management studies is the company, and the unit in many studies of crime rates is the city or municipality. In this context, "unit selection" could be assumed to involve selecting the unit of analysis which is different from our use of the term in this paper.

This leads us to objective functions of the following form:[4]

$$L(\mathbf{u}) = \sum_{i=1}^{n} w_i \cdot Pr(\mathbf{y}^i_{\mathbf{x}^i}, \mathbf{w}^i_{\mathbf{v}^i} | \mathbf{e}^i, \mathbf{u}) \quad \text{where } w_i \geq 0 \text{ and } \sum_{i=1}^{n} w_i = 1 \tag{1}$$

We can now formally define the unit selection inference problem on structural causal models.

Definition 1 (Unit Selection) *Given an SCM G, a subset $\mathbf{U}$ of its variables, and an objective function $L(\mathbf{u})$ such as Equation (1), the unit selection inference problem is to compute* $\operatorname{argmax}_{\mathbf{u}} L(\mathbf{u})$.

The benefit function discussed in (Li and Pearl, 2019) has the following form:

$$L(u) = \beta Pr(y_x, y'_{x'}|u) + \gamma Pr(y_x, y_{x'}|u) + \theta Pr(y'_x, y'_{x'}|u) + \delta Pr(y'_x, y_{x'}|u) \tag{2}$$

This class of objective functions falls as a special case of Equation (1) by setting $n = 4$, $\mathbf{E}^i = \emptyset$, $\mathbf{X}^i = \mathbf{V}^i = \{X\}$ and $\mathbf{Y}^i = \mathbf{W}^i = \{Y\}$ for $i = 1, \ldots, 4$, where X, Y are binary variables. That is, each component i of the objective function uses the same single, treatment variable X and the same single, outcome variable Y. A more general form was proposed in (Li and Pearl, 2022b) in which treatment X has values $x_1, \ldots, x_m$ and outcome Y has values $y_1, \ldots, y_k$ so the objective function can have up to k^m components, each corresponding to a distinct response type such as $Pr(y_{2x_1}, y_{1x_2}, y_{1x_3}, y_{3x_4}, y_{2x_5}|u)$ when $k = 3$ and $m = 5$. This class of objective functions is more general than Equation (1) in that it allows one to express more response types but it assumes one treatment variable and one outcome variable. The class of objective functions we consider in Equation (1) allows compound treatments and outcomes. It also allows us to seek units from a particular group. For example, if A and B are two medications (binary treatments) and T and P refer to high temperature and high blood pressure (binary outcomes), and E is the age group with values $e_1, \ldots, e_4$, then the objective function can include terms such as $Pr((t, p')_{a,b}\ ,\ (t', p')_{a',b} \mid e_3, u)$, which is the probability that a member of the third age group would have high temperature and normal blood pressure if administered both medications and would have normal temperature and blood pressure if administered only the second medication. Moreover, since the objective function components can have different treatment and outcome variables, one can select units based on their responses to distinct stimuli (e.g., effect of one type of encouragement on membership renewal and the simultaneous effect of another type of encouragement on increased purchases).[5]

4. The Complexity of Unit Selection

We show next that unit selection is NP^{PP}-complete for the class of causal objective functions given in Equation (1). We also show that this problem is NP-complete when unit variables correspond to all exogenous variables in the SCM.[6] We start by providing an efficient reduction from unit selection

4. The conditions we place on weights w_i are meant for convenience and they are not restrictive.
5. Going beyond the form in Equation (1), one can use causal objective functions with more general ingredients, such as: the probability of a patient being a responder given they are not a contrarian, $Pr(y_x, y'_{x'}|\neg(y'_x, y_{x'}))$; or the probability that a patient would not have had a stroke if they were on a diet (y'_d) or had exercised (y'_e) given that they did neither (d', e'), i.e., $Pr(y'_d \vee y'_e|y, d', e')$. Such general quantities have not been treated in the literature but some discussions have argued for their significance and treated some special cases; e.g., (Pearl, 2017).
6. For a discussion of complexity classes that are relevant to Bayesian network inference, see (Shimony, 1994) on the MPE decision problem being NP-complete, and (Park, 2002; Park and Darwiche, 2004a) on the MAP decision problem being NP^{PP}-complete. Roth (1996) shows that computing node marginals in a Bayesian network is #P-complete. For a textbook discussion of these complexity results, see (Darwiche, 2009, Ch. 11).

into a variant of the well-known MAP inference problem, which we call Reverse-MAP. We then follow by studying the complexity of Reverse-MAP and unit selection.

Recall that our goal is to find units $\mathbf{u}$ that maximize the value $L(\mathbf{u})$ of the objective function. The first step in solving this optimization problem is to be able to evaluate the objective $L(\mathbf{u})$. We next show a construction which allows us to evaluate $L(\mathbf{u})$ by evaluating a single observational probability involving unit variables $\mathbf{U}$ but on an extended and mutilated model. This construction will serve two purposes. First, it will permit us to characterize the complexity of unit selection when using objective functions in the form of Equation (1). Second, we will later use the construction to develop a specific algorithm for solving the unit selection problem using these objective functions.

Consider each term $Pr(\mathbf{y}^i_{\mathbf{x}^i}, \mathbf{w}^i_{\mathbf{v}^i}|\mathbf{e}^i, \mathbf{u})$ in Equation (1). We reviewed in Section 2 how this quantity can be reduced to a classical conditional probability on a triplet model G^i. The next step is to encode a linear combination of these conditional probabilities as a conditional probability on some model G'. This is done using the following construction.

Definition 2 (Objective Model) *Consider an SCM G with parameters θ and the objective function L in Equation (1). The objective model G' for $\langle G, L\rangle$ has parameters θ' and constructed as follows:*

1. *Construct a triplet model G^i of G for each term $Pr(\mathbf{y}^i_{\mathbf{x}^i}, \mathbf{w}^i_{\mathbf{v}^i}|\mathbf{e}^i, \mathbf{u})$ in L (see Section 2). Join $G^1, \ldots, G^n$ so that their unit variables $\mathbf{U}$ are shared. This leads to model G'.*

2. *Add a node H to G' as a parent of all outcome nodes $\mathbf{Z} = \{[\mathbf{Y}^i], [[\mathbf{W}^i]]\}_{i=1}^n$. Node H has states $h_1, \ldots, h_n$ and prior $\theta'(h_i) = w_i$. Each node $Z \in \mathbf{Z}$ now has parents $\mathbf{P}_Z \cup \{H\}$, where $\mathbf{P}_Z$ are the parents of Z in G' before node H is added. Let z^i be the state of Z in the corresponding instantiation $\mathbf{y}^i\mathbf{w}^i$ of objective function L. The new CPT for Z is:*

$\mathbf{P}_Z$	H	Z	$\theta'(Z\|\mathbf{P}_Z, H)$
$\mathbf{p}$	h_i	z^i	$\theta(z^i\|\mathbf{p})$
$\mathbf{p}$	h_i	$\bar{z}^i$	$\theta(\bar{z}^i\|\mathbf{p})$
$\mathbf{p}$	$\bar{h}_i$	z^i	1.0
$\mathbf{p}$	$\bar{h}_i$	$\bar{z}^i$	0.0

Here, $\bar{z}^i, \bar{h}_i$ denote any states of variables Z, H that are distinct from states z^i, h_i.

We say the objective model G' has n *components,* and call H the *mixture variable* as it encodes a mixture of the objective function terms. The CPTs for variables $[\mathbf{Y}^i], [[\mathbf{W}^i]]$ in model G' reduce to their original CPTs in SCM G when $H = h_i$, and imply $[\mathbf{Y}^i] = \mathbf{y}^i, [[\mathbf{W}^i]] = \mathbf{w}^i$ when $H \neq h_i$. The objective $L(\mathbf{u})$ in SCM G is a classical probability in the objective model G' (proof in Appendix A).

Theorem 3 *Consider an SCM G with unit variables $\mathbf{U}$. Let L be the objective function in Equation (1), and let G' be an objective model for $\langle G, L\rangle$. Let $\mathbf{X} = \{[\mathbf{X}^i]\}_{i=1}^n, \mathbf{Y} = \{[\mathbf{Y}^i]\}_{i=1}^n$, $\mathbf{W} = \{[[\mathbf{W}^i]]\}_{i=1}^n$, $\mathbf{V} = \{[[\mathbf{V}^i]]\}_{i=1}^n$ and $\mathbf{E} = \{\mathbf{E}^i\}_{i=1}^n$. We have $L(\mathbf{u}) = Pr'(\mathbf{y}, \mathbf{w}|\mathbf{x}, \mathbf{v}, \mathbf{e}, \mathbf{u})$, where $\mathbf{y}, \mathbf{w}, \mathbf{x}, \mathbf{v}, \mathbf{e}$ are the instantiations of variables $\mathbf{Y}, \mathbf{W}, \mathbf{X}, \mathbf{V}, \mathbf{E}$ in objective function L.*

Consider the SCM in Figure 1(*a*) and the causal objective function $L(u) = w_1 \cdot Pr(y_x, y'_{x'}|u) + w_2 \cdot Pr(y_x, y_{x'}|u)$. Figure 2 shows a corresponding objective model G' constructed according to Definition 2. We now have $L(u) = Pr'([y^1], [[y'^1]], [y^2], [[y^2]] \mid [x^1], [[x'^1]], [x^2], [[x'^2]], u)$.

Theorem 3 suggests that we can optimize the objective function $L(\mathbf{u})$ on an SCM G by computing the instantiation $\operatorname{argmax}_{\mathbf{u}} Pr(\mathbf{y}, \mathbf{w}|\mathbf{x}, \mathbf{v}, \mathbf{e}, \mathbf{u})$ on an objective model G'. The is similar to the classical MAP problem on model G', except that the optimized variables $\mathbf{U}$ appear after the conditioning operator instead of before it. This leads to our definition of the Reverse-MAP problem.

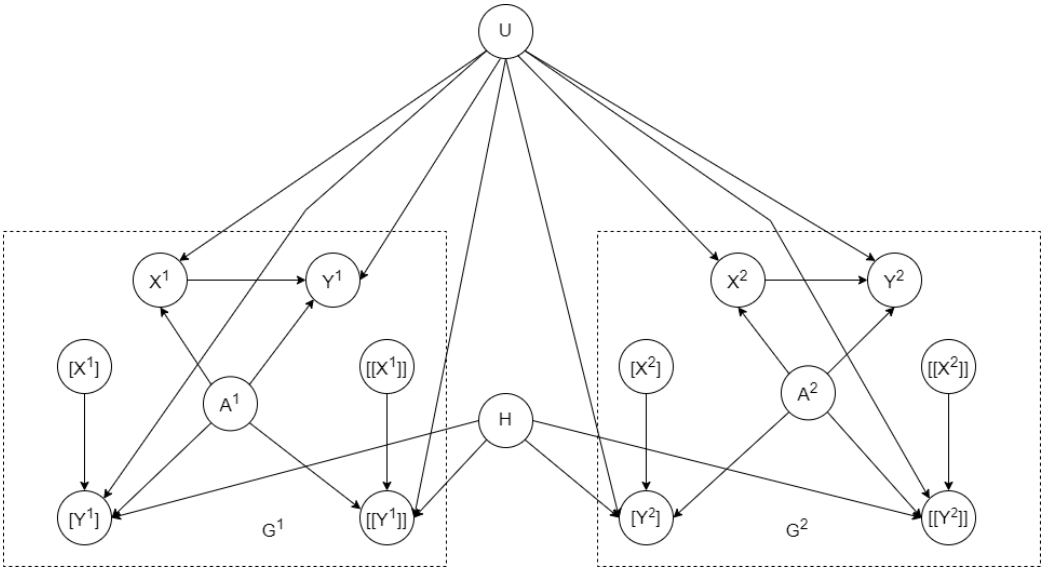


Figure 2: An objective model with two components for the SCM in Figure 1(*a*).

Definition 4 (Reverse-MAP) *Consider an SCM G with distribution Pr and suppose $\mathbf{U}, \mathbf{E}_1, \mathbf{E}_2$ are disjoint sets of variables in G. The Reverse-MAP instantiation for variables $\mathbf{U}$ and instantiations $\mathbf{e}_1, \mathbf{e}_2$ is defined as follows:* $\text{RMAP}(\mathbf{U}, \mathbf{e}_1, \mathbf{e}_2) \triangleq \text{argmax}_{\mathbf{u}} \, Pr(\mathbf{e}_1 \mid \mathbf{u}, \mathbf{e}_2)$.

To see the connection between Reverse-MAP and MAP, note that $\text{argmax}_{\mathbf{u}} \, Pr(\mathbf{e}_1|\mathbf{u}, \mathbf{e}_2) = \text{argmax}_{\mathbf{u}} \, Pr(\mathbf{u}, \mathbf{e}_1, \mathbf{e}_2)/Pr(\mathbf{u}, \mathbf{e}_2)$ where $\text{argmax}_{\mathbf{u}} \, Pr(\mathbf{u}, \mathbf{e}_1, \mathbf{e}_2) = \text{argmax}_{\mathbf{u}} \, Pr(\mathbf{u}|\mathbf{e}_1, \mathbf{e}_2)$ is the known MAP problem (Pearl, 1989). In general, the MAP instantiation $\text{argmax}_{\mathbf{u}} \, Pr(\mathbf{u}, \mathbf{e}_1, \mathbf{e}_2)$ is not the Reverse-MAP instantiation since $Pr(\mathbf{u}, \mathbf{e}_2)$ also depends on $\mathbf{U}$; see Appendix B for a concrete example that illustrates this point. We now have the following result, proven in Appendix C.

Corollary 5 *There are polynomial-time reductions between the Reverse-MAP problem and the unit selection problem with objective functions in the form of Equation (1).*

We next characterize the complexity of Reverse-MAP under different conditions. Consider a decision version of the problem, D-Reverse-MAP, defined as follows.

Definition 6 (D-Reverse-MAP) *Given an SCM with rational parameters that induces distribution Pr, some target variables $\mathbf{U}$, some evidence $\mathbf{e}_1, \mathbf{e}_2$ and a rational threshold p, the D-Reverse-MAP problems asks whether there is an instantiation $\mathbf{u}$ of $\mathbf{U}$ such that $Pr(\mathbf{e}_1|\mathbf{u}, \mathbf{e}_2) > p$.*

The next theorem shows that D-Reverse-MAP is NP^{PP}-complete, like classical MAP (Park and Darwiche, 2004b). Its proof can be found in Appendix D.

Theorem 7 *D-Reverse-MAP is NP^{PP}-complete.*

We can now characterize the complexity of unit selection using Theorem 7 and Corollary 5.

Corollary 8 *Unit selection is NP^{PP}-complete assuming the objective function in Equation (1).*

In an SCM, exogenous (root) variables represent all uncertainties in the model and the endogenous (internal) variables are uniquely determined by exogenous variables. This property of SCMs significantly reduces the complexity of unit selection when the unit variables correspond to all SCM exogenous variables. This is implied by the following result which is proven in Appendix E.

Theorem 9 *D-Reverse-MAP is NP-complete if its target variables are all the SCM root variables.*

Corollary 10 *Unit selection is NP-complete when the unit variables are all the SCM exogenous (root) variables, assuming the objective functions in Equation (1).*

5. Unit Selection using Variable Elimination

Section 4 provided a reduction from unit selection on an SCM to Reverse-MAP on an objective model. In Section 5.1, we provide a variable elimination (VE) algorithm for Reverse-MAP which can be applied to the objective model to solve unit selection. In Section 5.2, we analyze the complexity of this method and compare it to the complexity of Reverse-MAP on the underlying SCM.

5.1. Reverse-MAP using Variable Elimination

Our VE algorithm for Reverse-MAP will employ the same machinery and techniques used in the VE algorithm for classical MAP (Dechter, 1999). Hence, we will first review the VE algorithm for MAP using the treatment in (Darwiche, 2009, Ch 10) and then discuss the algorithm for Reverse-MAP.

The VE algorithm is based on the notion of a *factor* $f(\mathbf{X})$ which maps each instantiation $\mathbf{x}$ of variables $\mathbf{X}$ into a non-negative number $f(\mathbf{x})$. VE employs a number of factor operations including multiplying two factors ($f \cdot g$), summing out a variable from a factor ($\sum_X f$), maximizing out a variable from a factor ($\max_X f$), and dividing two factors (f/g). Let G be an SCM and assume its variables $\mathbf{Z}$ are partitioned into three disjoint sets $\mathbf{U}, \mathbf{V}, \mathbf{E}$, where $\mathbf{U}$ are the *target variables* and $\mathbf{E}$ are the *evidence variables*. Let $\mathbf{S} = \mathbf{Z} \setminus \mathbf{U}$ in the following discussion. We will treat the CPT of each variable Z in SCM G as a factor over Z and its parents $\mathbf{P}$, denoted $f_Z(Z\mathbf{P})$. The SCM distribution is then $Pr(\mathbf{Z}) = \prod_{Z \in \mathbf{Z}} f_Z$. We capture evidence $\mathbf{e}$ by creating an evidence factor $\lambda_e(E)$ for each $e \in \mathbf{e}$ with $\lambda_e(e') = 1$ if $e' = e$ and $\lambda_e(e') = 0$ otherwise. The *MAP probability* is then given by[7]

$$\text{MAP}_p(\mathbf{U}, \mathbf{e}) = \max_{\mathbf{u}} Pr(\mathbf{u}, \mathbf{e}) = \max_{\mathbf{u}} \sum_{\mathbf{v}} Pr(\mathbf{u}, \mathbf{v}, \mathbf{e}) = \max_{\mathbf{U}} \sum_{\mathbf{S}} \prod_{Z \in \mathbf{Z}} f_Z \prod_{e \in \mathbf{e}} \lambda_e(E) \quad (3)$$

This is in contrast to the *MAP instantiation* which is $\text{argmax}_{\mathbf{u}} Pr(\mathbf{u}, \mathbf{e})$. With some minor bookkeeping, the VE algorithm for computing the MAP probability can also return a MAP instantiation; see, e.g., (Darwiche, 2009, Ch 10). Hence, we will focus next on computing the MAP probability.

Consider the SCM in Figure 3 and suppose $\mathbf{U} = \{A, B\}$ and the evidence $\mathbf{e}$ is $\{E = e\}$. In this case, $\text{MAP}_p(AB, e)$ will be equal to

$$\max_{AB} \sum_{CDE} f_A(A) f_B(AB) f_C(AC) f_D(BCD) f_E(CE) \lambda_e(E) \quad (4)$$

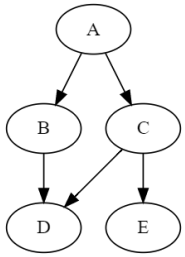


Figure 3: SCM

7. The left side of Equation 3 is a scalar (probability) while the right side is a factor over an empty set of variables, which is called a *scalar factor*. Such a factor maps only one instantiation, the empty one, to a scalar.

Algorithm 1 MAP_VE($G, \mathbf{U}, \mathbf{e}$)

Input: SCM G, target variables $\mathbf{U}$, evidence $\mathbf{e}$
Output: scalar factor containing $\text{MAP}_p(\mathbf{U}, \mathbf{e})$

1: **procedure** MAIN
2: $\pi_{\mathbf{S}} \leftarrow$ an elimination order for non-target variables
3: $\pi_{\mathbf{U}} \leftarrow$ an elimination order for target variables $\mathbf{U}$
4: $\mathcal{F} \leftarrow \{f : f \text{ is a CPT of SCM } G\} \cup \{\lambda_e : \lambda_e \text{ is an evidence factor for } e \in \mathbf{e}\}$
5: $\mathcal{G} \leftarrow$ ELIMINATE($\sum, \mathcal{F}, \pi_{\mathbf{S}}$)
6: $p \leftarrow$ ELIMINATE($\max, \mathcal{G}, \pi_{\mathbf{U}}$)
7: **return** p
8: **end procedure**

Algorithm 2 Eliminating Variables using Sum or Max

Input: an operation $\bigcirc \in \{\sum, \max\}$, a set of factors $\mathcal{F}$, a total variable order π
Output: a set of factors

1: **procedure** ELIMINATE($\bigcirc, \mathcal{F}, \pi$)
2: **for** $i = 1$ to length of order π **do**
3: $V \leftarrow i^{th}$ variable in order π
4: $\mathcal{G} \leftarrow$ factors in $\mathcal{F}$ that mention variable V
5: $f_i \leftarrow \prod_{f \in \mathcal{G}} f$
6: $f_i \leftarrow \bigcirc_V f_i$
7: replace factors $\mathcal{G}$ in $\mathcal{F}$ with factor f_i
8: **end for**
9: **return** $\mathcal{F}$
10: **end procedure**

A naive evaluation of this expression multiplies all factors to yield a factor $f(ABCDE)$ over all variables, then computes $\max_{AB} \sum_{CDE} f(ABCDE)$, leading to $O(n \exp(n))$ complexity where n is the number of model variables. The VE algorithm tries to compute this expression more efficiently with pseudocode provided in Algorithm 1 (MAP_VE). The product of factors $\mathcal{F}$ on Line 4 represents the joint distribution $Pr(\mathbf{Z}, \mathbf{e})$ so we first sum out variables $\mathbf{S}$ from $\mathcal{F}$ on Line 5 to compute a set of factors $\mathcal{G}$ whose product represents the marginal $Pr(\mathbf{U}, \mathbf{e})$. We then maximize out variables $\mathbf{U}$ from $\mathcal{G}$ on Line 6 leading to a scalar factor p that contains the MAP probability (see Footnote 7). Algorithm 1 eliminates variables one by one using Algorithm 2 and a total variable order $\pi = \langle \pi_{\mathbf{S}}, \pi_{\mathbf{U}} \rangle$, known as an *elimination order*. MAP_VE requires variables $\mathbf{U}$ to appear last in order π since summation does not commute with maximization. An order that satisfies this constraint is known as a $\mathbf{U}$-*constrained elimination order*. The complexity of MAP_VE depends on the used elimination order π. In each elimination step of Algorithm 2, we multiply all factors that mention variable $\pi(i)$ to obtain factor f_i on Line 6. The variables in f_i are called a cluster $\mathbf{C}_i$ so eliminating variables $\pi(1), \ldots, \pi(n)$ induces clusters $\mathbf{C}_1, \ldots, \mathbf{C}_n$. The *width* w of elimination order π is the size of largest cluster minus one and the complexity of MAP_VE is $O(n \exp(w))$.

The table below depicts the trace of MAP_VE when computing the MAP probability in Equation (4) using the elimination order $\pi = E, D, C, B, A$. The trace shows that MAP_VE evaluates

the following factorized expression and that the width of order π is 2 (largest cluster has size 3):

$$\text{MAP}_p(AB, e) = \max_A f_A(A) \left[\max_B f_B(AB) \left[\sum_C f_C(AC) \left[\sum_D f_D(BCD) \left[\sum_E f_E(CE)\lambda_E \right] \right] \right] \right]$$

i	eliminated var	factors $\mathcal{G}$ (Line 4, Algorithm 2)	new factor f_i (Line 6, Algorithm 2)	$\mathbf{C}_i$
1	E	$f_E(CE)\ \lambda_E$	$f_1 = \sum_E f_E(CE)\ \lambda_E$	CE
2	D	$f_D(BCD)$	$f_2 = \sum_D f_D(BCD)$	BCD
3	C	$f_C(AC)\ f_1(C)\ f_2(BC)$	$f_3 = \sum_C f_C(AC)\ f_1(C)\ f_2(BC)$	ABC
4	B	$f_B(AB)\ f_3(AB)$	$f_4 = \max_B f_B(AB)\ f_3(AB)$	AB
5	A	$f_A(A)\ f_4(A)$	$p = \max_A f_A(A)\ f_4(A)$	A

Choosing a good elimination order is critical for the complexity of VE. The *treewidth* of an SCM G is defined as the minimum width attained by any elimination order. Since MAP requires $\mathbf{U}$-constrained orders, the $\mathbf{U}$-*constrained treewidth* of G is defined as the minimum width attained by any $\mathbf{U}$-constrained elimination order (Park and Darwiche, 2004b).

We are now ready to introduce our VE algorithm for Reverse-MAP. Again, we assume that the model variables $\mathbf{Z}$ are partitioned into disjoint sets $\mathbf{U}, \mathbf{V}, \mathbf{E}$, where $\mathbf{U}$ are the target variables and $\mathbf{S} = \mathbf{Z} \setminus \mathbf{U}$. But we further partition the evidence variables $\mathbf{E}$ into $\mathbf{E}_1$ and $\mathbf{E}_2$. Again, we focus on computing the Reverse-MAP probability $\text{RMAP}_p(\mathbf{U}, \mathbf{e}_1, \mathbf{e}_2)$ instead of the instantiation:

$$\max_{\mathbf{u}} Pr(\mathbf{e}_1|\mathbf{u}, \mathbf{e}_2) = \max_{\mathbf{u}} \frac{Pr(\mathbf{u}, \mathbf{e}_1, \mathbf{e}_2)}{Pr(\mathbf{u}, \mathbf{e}_2)} = \max_{\mathbf{u}} \frac{\sum_{\mathbf{v}} Pr(\mathbf{u}, \mathbf{v}, \mathbf{e}_1, \mathbf{e}_2)}{\sum_{\mathbf{v}} Pr(\mathbf{u}, \mathbf{v}, \mathbf{e}_2)} = \max_{\mathbf{U}} \frac{\sum_{\mathbf{S}} \prod_{Z \in \mathbf{Z}} f_Z \prod_{e \in \mathbf{e}_1 \cup \mathbf{e}_2} \lambda_e}{\sum_{\mathbf{S}} \prod_{Z \in \mathbf{Z}} f_Z \prod_{e \in \mathbf{e}_2} \lambda_e}$$

Our algorithm, called RMAP_VE, runs two passes of elimination as shown in Algorithm 3. In the first pass (Line 4), we sum out variables $\mathbf{S}$ under evidence $\mathbf{e}_1, \mathbf{e}_2$ and in the second pass (Line 5), we sum out variables $\mathbf{S}$ under evidence $\mathbf{e}_2$. This leads to two sets of factors $\mathcal{G}_1$ and $\mathcal{G}_2$ which correspond to marginal distributions $Pr(\mathbf{U}, \mathbf{e}_1, \mathbf{e}_2)$ and $Pr(\mathbf{U}, \mathbf{e}_2)$. Now we need to divide $Pr(\mathbf{U}, \mathbf{e}_1, \mathbf{e}_2)$ and $Pr(\mathbf{U}, \mathbf{e}_2)$ to compute $Pr(\mathbf{e}_1|\mathbf{U}, \mathbf{e}_2)$. We next show that this can be done efficiently by "dividing" $\mathcal{G}_1$ and $\mathcal{G}_2$ as shown on Line 8. The key idea is that if we run the two passes of elimination according to the same elimination order, then there will be a one-to-one correspondence between the factors in $\mathcal{G}_1$ and $\mathcal{G}_2$. Let (g_1^i, g_2^i) be the corresponding pairs of factors for $i = 1, \ldots, k$ where $k = |\mathcal{G}_1| = |\mathcal{G}_2|$. What we need is $\left(\prod_{i=1}^n g_1^i\right) / \left(\prod_{i=1}^n g_2^i\right)$ since this represents $Pr(\mathbf{e}_1|\mathbf{U}, \mathbf{e}_2)$. But due to the mentioned correspondence, this equals $\prod_{i=1}^n g_1^i / g_2^i$. Thus, we can divide each pair of corresponding factors to obtain the set of factors $\mathcal{G}$ as done on Line 8. We finally maximize out target variables $\mathbf{U}$ from $\mathcal{G}$ to obtain the Reverse-MAP probability (Line 9).

RMAP_VE has the same complexity as MAP_VE if both use the same elimination order. Suppose there are k factors in $\mathcal{G}_1/\mathcal{G}_2/\mathcal{G}$ and the largest factor has size c. The cost of division on Line 8 is $O(k \exp(c))$ while the cost of maximization on Line 9 is at least $O(k \exp(c))$ so the cost of division is dominated by the cost of maximization. Hence, the complexity of RMAP_VE is still $O(n \exp(w))$ where n is the number of variables and w is the width of used $\mathbf{U}$-constrained order π.

Algorithm 3 RMAP_VE($G, \mathbf{U}, \mathbf{e}_1, \mathbf{e}_2$)

Input: SCM G, target variables $\mathbf{U}$, evidence $\mathbf{e}_1$ and $\mathbf{e}_2$
Output: scalar factor containing $\text{RMAP}_p(\mathbf{U}, \mathbf{e}_1, \mathbf{e}_2)$

procedure MAIN
  $\pi_{\mathbf{S}} \leftarrow$ an elimination order for non-target variables
  $\pi_{\mathbf{U}} \leftarrow$ an elimination order for target variables $\mathbf{U}$
  $\mathcal{F}_1 \leftarrow \{f : f \text{ is a CPT of SCM } G\} \cup \{\lambda_e : \lambda_e \text{ is an evidence factor for } e \in \mathbf{e}_1, \mathbf{e}_2\}$
  $\mathcal{F}_2 \leftarrow \{f : f \text{ is a CPT of SCM } G\} \cup \{\lambda_e : \lambda_e \text{ is an evidence factor for } e \in \mathbf{e}_2\}$
  $\mathcal{G}_1 \leftarrow \text{ELIMINATE}(\sum, \mathcal{F}_1, \pi_{\mathbf{S}})$
  $\mathcal{G}_2 \leftarrow \text{ELIMINATE}(\sum, \mathcal{F}_2, \pi_{\mathbf{S}})$
  $\mathcal{G} \leftarrow \{g_1/g_2 : g_1, g_2 \text{ are corresponding factors in } \mathcal{G}_1, \mathcal{G}_2\}$
  $p \leftarrow \text{ELIMINATE}(\max, \mathcal{G}, \pi_{\mathbf{U}})$
  return p
end procedure

5.2. Bounding the Complexity of Unit Selection using Variable Elimination

We can solve unit selection by applying RMAP_VE to an objective model of the SCM as shown by Theorem 3. However, RMAP_VE (and MAP_VE) is expected to be more expensive on the objective model compared to the given SCM since the former is larger and denser than the latter. But how much more expensive? In particular, is RMAP_VE always tractable on the objective model when it is tractable on the underlying SCM? We consider this question next using the lens of treewidth which is commonly used to analyze elimination algorithms. Recall also that MAP_VE and RMAP_VE have the same complexity when applied to the same SCM using the same target variables.

Our starting point is to study the treewidth of an objective model in relation to the treewidth of its underlying SCM. We will base our study on the techniques and results reported in (Han et al., 2022) which studied the complexity of counterfactual reasoning. In particular, given an elimination order π of SCM G, we next show how to construct an elimination order π' for the objective model G' while providing a bound on the width of order π' in terms of the width of order π. Recall that we use $[X]$ and $[[X]]$ to denote the copies of variable X in a triplet model where $X = [X] = [[X]]$ if X is exogenous. Moreover, if U is a unit variable, then $U = U^1 = \cdots = U^n$ in an objective model.

Definition 11 *Let G be an SCM and G' be a corresponding objective model with n components. If π is an elimination order for G, the corresponding elimination order π' for G' is obtained by replacing each non-unit variable X in π by $X^1, \ldots, X^n, [X^1], \ldots, [X^n], [[X^1]], \ldots, [[X^n]]$ then appending the mixture variable H to the end of the order.*

Consider the elimination order $\pi = A, X, Y, U$ for the SCM in Figure 1(*a*). The corresponding elimination order π' for the objective model in Figure 2 is as follows:

$$\pi' = A^1, A^2, X^1, X^2, [X^1], [X^2], [[X^1]], [[X^2]], Y_1, Y_2, [Y_1], [Y_2], [[Y_1]], [[Y_2]], U, H$$

The following bound (Theorem 14) follows from Lemma 12 and Theorem 13 which concerns *n-world models*. Given an SCM G and a subset $\mathbf{U}$ of its roots, an n-world model is obtained by creating n copies of G that share nodes $\mathbf{U}$ (Han et al., 2022). This notion corresponds to parallel worlds models (Avin et al., 2005) when $\mathbf{U}$ contains all roots of SCM G. An objective model with n components can be viewed as a $3n$-world model but with an additional mixture node H and

some edges that originate from H. Lemma 12 and Theorem 14 are proven in Appendix G and Appendix H.

Lemma 12 *Consider an SCM G and suppose SCM G' is obtained from G by adding a root node H as a parent of some nodes in G. If π is an elimination order for G and has width w, then $\pi' = \langle \pi, H \rangle$ is an elimination order for G' and has width $w' \leq w + 1$.*

Theorem 13 (Han et al. (2022)) *Consider an SCM G, a subset $\mathbf{U}$ of its roots and a corresponding n-world model G'. If G has an elimination order π with width w, then there exists a corresponding elimination order π' of G' that has width $w' \leq n(w+1) - 1$.*

Theorem 14 *Consider an SCM G and a corresponding objective model G' with n components. Let π be an elimination order for G and let π' be the corresponding elimination order for G'. If π has width w and π' has width w', then $w' \leq 3n(w+1)$.*

Corollary 15 *If w is the treewidth of an SCM G and w' is the treewidth of a corresponding objective model G' with n components, then $w' \leq 3n(w+1)$.*

As mentioned earlier, RMAP_VE and MAP_VE require a $\mathbf{U}$-constrained elimination orders in which unit variables $\mathbf{U}$ appear last in the order. Hence, a $\mathbf{U}$-constrained elimination order for an objective model must place the mixture variable H before $\mathbf{U}$. This leads to the next definition.

Definition 16 *Let G be an SCM with unit variables $\mathbf{U}$ and let G' be a corresponding objective model with n components. If π is a $\mathbf{U}$-constrained elimination order for G, the corresponding $\mathbf{U}$-constrained elimination order π' for G' is obtained by replacing each non-unit variable X in π by $X^1, \ldots, X^n, [X^1], \ldots, [X^n], [[X^1]], \ldots, [[X^n]]$ then inserting mixture variable H just before $\mathbf{U}$.*

Consider the $\mathbf{U}$-constrained order $\pi = A, X, Y, U$ for the SCM in Figure 1(*a*). The corresponding $\mathbf{U}$-constrained elimination order for the objective model in Figure 2 is

$$\pi' = A^1, A^2, X^1, X^2, [X^1], [X^2], [[X^1]], [[X^2]], Y_1, Y_2, [Y_1], [Y_2], [[Y_1]], [[Y_2]], H, U$$

We now have the following bound on the $\mathbf{U}$-constrained treewidth of objective models, which is somewhat unexpected when compared to the bound on treewidth. In particular, while the bound on treewidth grows linearly in the number of components in the objective model, the bound on $\mathbf{U}$-constrained treewidth is independent of such a number. Moreover, the bound on $\mathbf{U}$-constrained treewidth can depend on the number of unit variables which is not the case for treewidth.

Theorem 17 *Let G be an SCM with unit variables $\mathbf{U}$ and let G' be a corresponding objective model. If π is a $\mathbf{U}$-constrained elimination order for G with width w and π' is the corresponding $\mathbf{U}$-constrained elimination order for G' with width w', then $w' \leq \max(3w+3, |\mathbf{U}|)$. If the objective function in Equation (1) has one outcome variable ($\mathbf{Y}^i = \mathbf{W}^i = \{Y\}$ for all i), then $w' \leq 3w+3$.*

Corollary 18 *Let G be an SCM with unit variables $\mathbf{U}$ and let G' be a corresponding objective model. If w and w' are the $\mathbf{U}$-constrained treewidths of G and G', then $w' \leq \max(3w+3, |\mathbf{U}|)$. Moreover, if the objective function in Equation (1) has a single outcome variable, then $w' \leq 3w+3$.*

The above bounds can be significantly tighter depending on the objective function properties. Corollary 18 identifies one such property which is satisfied by the benefit function in (Li and Pearl, 2019); see Equation (2). Moreover, the factor 3 in these bounds is an implication of using a triplet model which may not be necessary. Consider components $Pr(\mathbf{y}^i_{\mathbf{x}^i}, \mathbf{w}^i_{\mathbf{v}^i}|\mathbf{e}^i, \mathbf{u})$ in the objective function of Equation (1). If $\mathbf{E}^i{=}\emptyset$ for all i, then a twin model is sufficient when building an objective model (similarly if $\mathbf{Y}^i{=}\mathbf{X}^i{=}\emptyset$ or $\mathbf{W}^i{=}\mathbf{V}^i{=}\emptyset$). The objective function in Equation (2), from (Li and Pearl, 2019), has $\mathbf{E}^i{=}\emptyset$ for all i so it leads to the tighter bound $w' \leq 2w + 2$. More generally, if the objective function properties lead to removing the dependence on $|\mathbf{U}|$ in the bound of Corollary 18, then RMAP_VE on an objective model is tractable if RMAP_VE (MAP_VE) is tractable on the underlying SCM. Otherwise, the bound in Corollary 18 does not guarantee this. Recall that MAP, Reverse-MAP and unit selection using Equation (1) are all NP^{PP}-complete as shown earlier.

We provide in Appendix J a preliminary experiment and an extensive discussion in relation to the complexities of three algorithms: (1) MAP_VE (Algorithm 1) which solves MAP by operating on an SCM; (2) RMAP_VE (Algorithm 3) which solves unit selection by operating on an objective model; and (3) a baseline, bruteforce method which solves unit selection by operating on a twin or triplet model (depending on the objective function). The main finding of the experiment is that, as the size of the problem grows,[8] the gap between the complexities of MAP_VE and RMAP_VE narrows while the gap between the complexities of RMAP_VE and the bruteforce method grows (the bruteforce method is significantly worse and becomes impractical pretty quickly). Appendix J also identifies a class of SCM structures (and unit variables) for which the number of unit variables is unbounded but the complexity of RMAP_VE on an objective model is bounded.

We finally note that the complexity bounds we provided for MAP_VE and RMAP_VE are in terms of treewidth, but tighter bounds can be obtained using the recent notion of *causal treewidth* (Chen and Darwiche, 2022; Darwiche, 2020), which is no greater than treewidth and can sometimes be bounded when treewidth is not (Darwiche, 2022) by leveraging functional dependencies in the SCMs. This is a subject for future work.

6. Conclusion

We studied the unit selection problem in a computational setting which complements existing studies. We assumed a fully specified structural causal model so we can compute point values of causal objective functions, allowing us to entertain a broader class of functions than is normally considered. We showed that the unit selection problem with this class of objective functions is NP^{PP}-complete, similar to the classical MAP problem, and identified an intuitive condition under which it is NP-complete. We further provided an exact algorithm for the unit selection problem based on variable elimination and characterized its complexity in terms of treewidth, while relating this complexity to that of MAP inference. In the process, we defined a new inference problem, Reverse-MAP, which is also NP^{PP}-complete but captures the essence of unit selection more than MAP does.

Acknowledgments

We thank Yizuo Chen, Yunqiu Han, Ang Li and Scott Mueller for providing useful feedback on an earlier version of this paper. This work has been partially supported by ONR grant N000142212501.

8. The size of the problem is measured by the number of nodes in the SCM and the number of unit variables.

References

Chen Avin, Ilya Shpitser, and Judea Pearl. Identifiability of path-specific effects. In *IJCAI*, pages 357–363. Professional Book Center, 2005.

Alexander Balke and Judea Pearl. Probabilistic evaluation of counterfactual queries. In *AAAI*, pages 230–237. AAAI Press / The MIT Press, 1994.

Alexander Balke and Judea Pearl. Counterfactuals and policy analysis in structural models. In *UAI*, pages 11–18. Morgan Kaufmann, 1995.

E. Bareinboim, Juan David Correa, D. Ibeling, and Thomas F. Icard. On Pearl's hierarchy and the foundations of causal inference. 2021. Technical Report, R-60, Colombia University.

Yizuo Chen and Adnan Darwiche. On the definition and computation of causal treewidth. In *Uncertainty in Artificial Intelligence*, pages 368–377. PMLR, 2022.

Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

Adnan Darwiche. An advance on variable elimination with applications to tensor-based computation. In *ECAI 2020*, pages 2559–2568. IOS Press, 2020.

Adnan Darwiche. Causal inference using tractable circuits. *arXiv preprint arXiv:2202.02891*, 2022.

Philip Dawid, Monica Musio, and Rossella Murtas. The probability of causation. *Law, Probability and Risk*, (16):163–179, 2017.

Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artif. Intell.*, 113(1-2): 41–85, 1999.

David Galles and Judea Pearl. An axiomatic characterization of causal counterfactuals. *Foundations of Science*, 3(1):151–182, 1998.

Joseph Y Halpern. Axiomatizing causal reasoning. *Journal of Artificial Intelligence Research*, 12: 317–337, 2000.

Yunqiu Han, Yizuo Chen, and Adnan Darwiche. On the complexity of counterfactual reasoning. In *A causal view on dynamical systems, NeurIPS 2022 workshop*, 2022. URL https://arxiv.org/abs/2211.13447.

Uffe Bro Kjærulff. Triangulation of graphs-algorithms giving small total state space. 1990.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

A. Li, S. Jiang, Y. Sun, and J. Pearl. Learning probabilities of causation from finite population data. Technical Report R-519, <http://ftp.cs.ucla.edu/pub/stat_ser/r519.pdf>, Department of Computer Science, University of California, Los Angeles, CA, 2022a.

Ang Li and Judea Pearl. Unit selection based on counterfactual logic. In *IJCAI*, pages 1793–1799. ijcai.org, 2019.

Ang Li and Judea Pearl. Unit selection with causal diagram. In *AAAI*, pages 5765–5772. AAAI Press, 2022a.

Ang Li and Judea Pearl. Unit selection with nonbinary treatment and effect. *CoRR*, abs/2208.09569, 2022b.

Ang Li and Judea Pearl. Unit selection: Case study and comparison with A/B test heuristic. *CoRR*, abs/2210.05030, 2022c.

Ang Li, Song Jiang, Yizhou Sun, and Judea Pearl. Unit selection: Learning benefit function from finite population data. *CoRR*, abs/2210.08203, 2022b.

Michael L Littman, Judy Goldsmith, and Martin Mundhenk. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9:1–36, 1998.

S. Mueller, A. Li, and J. Pearl. Causes of effects: Learning individual responses from population data. Technical Report R-505, <http://ftp.cs.ucla.edu/pub/stat_ser/r505.pdf>, Department of Computer Science, University of California, Los Angeles, CA, 2021. Forthcoming, Proceedings of IJCAI-2022.

James D Park. Map complexity results and approximation methods. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 388–396, 2002.

James D. Park and Adnan Darwiche. Complexity results and approximation strategies for MAP explanations. *J. Artif. Intell. Res. (JAIR)*, 21:101–133, 2004a.

James D. Park and Adnan Darwiche. Complexity results and approximation strategies for MAP explanations. *J. Artif. Intell. Res.*, 21:101–133, 2004b.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. MK, 1988.

Judea Pearl. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1989.

Judea Pearl. *Causality*. Cambridge University Press, 2000.

Judea Pearl. Physical and metaphysical counterfactuals: Evaluating disjunctive actions. *Journal of Causal Inference*, 5(2), 2017. doi: https://doi.org/10.1515/jci-2017-0018.

Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, 2018.

Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19(2), 2000.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

Dan Roth. On the hardness of approximate reasoning. *Artif. Intell.*, 82(1-2):273–302, 1996.

Solomon Eyal Shimony. Finding maps for belief networks is np-hard. *Artificial intelligence*, 68(2): 399–410, 1994.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search, Second Edition*. Adaptive computation and machine learning. MIT Press, 2000.

Jin Tian and Judea Pearl. Probabilities of causation: Bounds and identification. *Ann. Math. Artif. Intell.*, 28(1-4):287–313, 2000.

Appendix A. Proof of Theorem 3

The proof of this theorem requires a lemma which requires the following definition. We will say that a set of variables $\mathbf{Z}$ *decomposes* a DAG if removing the outgoing edges from $\mathbf{Z}$ splits the DAG into at least two disconnected components.

Lemma 19 *Consider an SCM G with distribution Pr and three disjoint set of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$. Suppose $\mathbf{Z}$ decomposes G into disconnected components G_1 and G_2. If $\mathbf{X}_1, \mathbf{Y}_1$ are subsets of $\mathbf{X}, \mathbf{Y}$ pertaining to G_1, and $\mathbf{X}_2, \mathbf{Y}_2$ are subsets of $\mathbf{X}, \mathbf{Y}$ pertaining to G_2, then $Pr(\mathbf{y}|\mathbf{x}, \mathbf{z}) = Pr(\mathbf{y}_1|\mathbf{x}_1, \mathbf{z})Pr(\mathbf{y}_2|\mathbf{x}_2, \mathbf{z})$.*

Proof Since $\mathbf{Z}$ decomposes G, we have $\mathrm{dsep}_G(\mathbf{X}_1, \mathbf{Z}, \mathbf{X}_2)$ and $\mathrm{dsep}_G(\mathbf{X}_1\mathbf{Y}_1, \mathbf{Z}, \mathbf{X}_2\mathbf{Y}_2)$. We have:

$$\begin{aligned} Pr(\mathbf{y}|\mathbf{x}, \mathbf{z}) &= \frac{Pr(\mathbf{y}, \mathbf{x}|\mathbf{z})}{Pr(\mathbf{x}|\mathbf{z})} = \frac{Pr(\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1, \mathbf{x}_2|\mathbf{z})}{Pr(\mathbf{x}_1, \mathbf{x}_2|\mathbf{z})} = \frac{Pr(\mathbf{y}_1, \mathbf{x}_1|\mathbf{z})Pr(\mathbf{y}_1, \mathbf{x}_1|\mathbf{z})}{Pr(\mathbf{x}_1|\mathbf{z})Pr(\mathbf{x}_2|\mathbf{z})} \qquad (5) \\ &= \left[\frac{Pr(\mathbf{y}_1, \mathbf{x}_1|\mathbf{z})}{Pr(\mathbf{x}_1|\mathbf{z})}\right]\left[\frac{Pr(\mathbf{y}_2, \mathbf{x}_2|\mathbf{z})}{Pr(\mathbf{x}_2|\mathbf{z})}\right] = Pr(\mathbf{y}_1|\mathbf{x}_1, \mathbf{z})Pr(\mathbf{y}_2|\mathbf{x}_2, \mathbf{z}) \end{aligned}$$

Equation (5) follows from $\mathrm{dsep}_G(\mathbf{X}_1, \mathbf{Z}, \mathbf{X}_2)$ and $\mathrm{dsep}_G(\mathbf{X}_1\mathbf{Y}_1, \mathbf{Z}, \mathbf{X}_2\mathbf{Y}_2)$. This concludes our proof. Although we only consider the case of two subnetworks here, it is easy to see that this lemma generalizes to an arbitrary number of subnetworks decomposed by $\mathbf{Z}$. ■

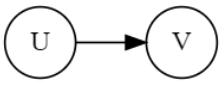
We are now ready to prove Theorem 3. By construction of G', $\mathbf{U} \cup \{H\}$ decomposes G' into its n components $G^1, G^2, \ldots G^n$. We have:

$$\begin{aligned} Pr'(\mathbf{y}, \mathbf{w} \mid \mathbf{x}, \mathbf{v}, \mathbf{e}, \mathbf{u}) &= \sum_{i=1}^{n} Pr'(\mathbf{y}, \mathbf{w}, h_i \mid \mathbf{x}, \mathbf{v}, \mathbf{e}, \mathbf{u}) \\ &= \sum_{i=1}^{n} Pr'(\mathbf{y}, \mathbf{w} \mid \mathbf{x}, \mathbf{v}, \mathbf{e}, \mathbf{u}, h_i) Pr'(h_i \mid \mathbf{x}, \mathbf{v}, \mathbf{e}, \mathbf{u}) \\ &= \sum_{i=1}^{n} Pr'(\mathbf{y}, \mathbf{w} \mid \mathbf{x}, \mathbf{v}, \mathbf{e}, \mathbf{u}, h_i) Pr'(h_i) \qquad (6) \\ &= \sum_{i=1}^{n} \left[\prod_{j=1}^{n} Pr'(\mathbf{y}^j, \mathbf{w}^j \mid \mathbf{x}^j, \mathbf{v}^j, \mathbf{e}^j, \mathbf{u}, h_i)\right] Pr'(h_i) \qquad (7) \\ &= \sum_{i=1}^{n} \left[\prod_{j\neq i} pr'(\mathbf{y}^j, \mathbf{w}^j \mid \mathbf{x}^j, \mathbf{v}^j, \mathbf{e}^j, \mathbf{u}, h_i)\right] Pr'(\mathbf{y}^i, \mathbf{w}^i \mid \mathbf{x}^i, \mathbf{v}^i, \mathbf{e}^i, \mathbf{u}, h_i) Pr'(h_i) \qquad (8) \\ &= \sum_{i=1}^{n} \left[\prod_{j\neq i} 1.0\right] Pr'(\mathbf{y}^i, \mathbf{w}^i \mid \mathbf{x}^i, \mathbf{v}^i, \mathbf{e}^i, \mathbf{u})\, w_i \qquad (9) \\ &= \sum_{i=1}^{n} w_i\, Pr(\mathbf{y}^i_{\mathbf{x}^i}, \mathbf{w}^i_{\mathbf{v}^i} \mid \mathbf{e}^i, \mathbf{u}) \\ &= L(\mathbf{u}) \qquad (10) \end{aligned}$$

Equation (6) follows since the auxiliary root H is d-separated from $\mathbf{X} \cup \mathbf{V} \cup \mathbf{E} \cup \mathbf{U}$. Equation (7) follows from Lemma 19 since $\mathbf{U} \cup \{H\}$ decomposes G such that all triplet models are disconnected. Equation (8) follows from the construction of the new CPTs of $\mathbf{Y}^i$ and $\mathbf{W}^i$: if $H = h_i$, then the original CPTs of $\mathbf{Y}^i$ and $\mathbf{W}^i$ are preserved, and the values of $\mathbf{Y}^j$ and $\mathbf{W}^j$ are fixed to $\mathbf{y}^j$ and $\mathbf{w}$ for all $j \neq i$. Equation (9) follow from the property of the triplet network.

Appendix B. Example for MAP and Reverse-MAP

Consider the simple model in Figure 4. We have $\text{argmax}_u \, Pr(u, v_1) = u_2$ for MAP while $\text{argmax}_u \, Pr(v_1|u) = u_1$ for R-MAP.



(a)

U	$\theta(A)$
u_1	0.2
u_2	0.8

(b)

U	V	$\theta(V\|U)$
u_1	v_1	0.6
u_1	v_2	0.4
u_2	v_1	0.3
u_2	v_2	0.7

(c)

Figure 4: An example for illustrating the difference between MAP and R-MAP.

Appendix C. Proof of Corollary 5

We can reduce Reverse-MAP $\text{argmax}_{\mathbf{u}} \, Pr(\mathbf{e}_1|\mathbf{u}, \mathbf{e}_2)$ to unit selection by choosing an objective function in the form of Equation (1) with the following settings: $n = 1$, $w_1 = 1$, $\mathbf{X}^1 = \{\}$, $\mathbf{Y}^1 = \mathbf{E}_1$ and $\mathbf{E}^1 = \mathbf{E}_2$. This is clearly a polynomial-time reduction. We already showed a reduction from unit selection to Reverse-MAP in Theorem 3. Let $|G|$ denote the size[9] of SCM G and n be the number of components in the causal objective function. By inspecting Definition 2, we can immediately see that the time for constructing the objective model G' is $O(n \cdot |G|)$. Moreover, the size of objective model G' is also $O(n \cdot |G|)$.

Appendix D. Proof of Theorem 7

Membership in NP^{PP} is immediate. Given an instantiation $\mathbf{u}$ of $\mathbf{U}$, it is easy to verify if $\mathbf{u}$ is a solution by querying the PP-oracle if $Pr(\mathbf{e}_1|\mathbf{u}, \mathbf{e}_2) > p$ which is a problem known as D-MAR (Darwiche, 2009). To prove hardness, we show that E-MAJSAT (Littman et al., 1998) can be reduced to D-Reverse-MAP in polynomial time, based on a slight modification of the reduction to classical MAP proposed in (Park, 2002; Park and Darwiche, 2004b). The E-MAJSAT problem is defined as follows. Given a Boolean formula α over Boolean variables $\mathbf{Z} = \mathbf{U} \cup \mathbf{V}$: Is there an instantiation $\mathbf{u}$ of $\mathbf{U}$ such that the majority of instantiations $\mathbf{v}$ of $\mathbf{V}$ satisfy $\mathbf{uv} \models \alpha$ (formula α holds at $\mathbf{uv}$)? We show that we can answer E-MAJSAT by answering D-Reverse-MAP on an SCM G_α that simulates the formula α and that can be constructed efficiently. The SCM G_α is constructed inductively, as

9. The size of an SCM is the space needed to store the SCM structure and parameters. For example, if the SCM is represented by a functional Bayesian Network, its size is usually the total number of entries in the network CPTs.

shown in (Park and Darwiche, 2004b),[10] and always has a single leaf node, denoted S_α. The construction is based on three rules: (1) If $\alpha = X$, then G_α has a single binary node X with values $\{0, 1\}$ and a uniform prior so $S_\alpha = X$; (2) If $\alpha = \neg\beta$, then G_α is constructed from G_β by adding a binary node S_α as a child of $S_\beta \in G_\beta$ with structural equation $S_\alpha = 1 - S_\beta$; and (3) If $\alpha = \beta \wedge \gamma$ ($\alpha = \beta \vee \gamma$), then G_α is constructed from G_β and G_γ by adding a binary node S_α as a child of $S_\beta \in G_\beta$ and $S_\gamma \in G_\gamma$ with structural equation $S_\alpha = S_\beta \cdot S_\gamma$ ($S_\alpha = S_\beta + S_\gamma$). We are now ready for the last step of the proof. Given a Boolean formula α over variables $\mathbf{Z} = \mathbf{U} \cup \mathbf{V}$, and given its SCM G_α that has distribution Pr, we next show that there is an instantiation $\mathbf{u}$ such that $Pr(S_\alpha = 1|\mathbf{u}) > 1/2$ (D-Reverse-MAP query) iff there is an instantiation $\mathbf{u}$ such that the majority of instantiations $\mathbf{v}$ of $\mathbf{V}$ satisfy $\mathbf{uv} \models \alpha$ (E-MAJSAT query). Let $|\mathbf{Z}| = n$ and s_α denote $S_\alpha = 1$. By construction of G_α (Park and Darwiche, 2004b), we have $Pr(\mathbf{z}) = 1/2^n$ for all instantiations $\mathbf{z}$; $Pr(s_\alpha|\mathbf{z}) = 1$ if $\mathbf{z} \models \alpha$ and $Pr(s_\alpha|\mathbf{z}) = 0$ otherwise. Then $Pr(\mathbf{z}, s_\alpha) = Pr(s_\alpha|\mathbf{z})Pr(\mathbf{z}) = 1/2^n$ if $\mathbf{z} \models \alpha$ and $Pr(\mathbf{z}, s_\alpha) = 0$ otherwise. We finally have:

$$Pr(s_\alpha|\mathbf{u}) = \frac{Pr(\mathbf{u}, s_\alpha)}{Pr(\mathbf{u})} = \frac{\sum_{\mathbf{v}} Pr(\mathbf{u}, \mathbf{v}, s_\alpha)}{\sum_{\mathbf{v}} Pr(\mathbf{u}, \mathbf{v})} = \frac{\sum_{\mathbf{v}:\mathbf{uv}\models\alpha}(1/2^n)}{\sum_{\mathbf{v}}(1/2^n)} = \frac{\mathbf{card}(\{\mathbf{v} \in \mathcal{V} : \mathbf{uv} \models \alpha\})}{\mathbf{card}(\mathcal{V})}$$

Now that we have shown membership and hardness, D-Reverse-MAP is NP^{PP}-complete.

Appendix E. Proof of Theorem 9

Recall Definition 6 of D-Reverse-MAP: Is there an instantiation $\mathbf{u}$ such that $Pr(\mathbf{e}_1|\mathbf{u}, \mathbf{e}_2) > p$? Membership in NP is immediate. Since $\mathbf{U}$ is the set of exogenous variables, evidence variables $\mathbf{E}$ are functionally determined by $\mathbf{U}$. Hence, it is easy to check whether an instantiation $\mathbf{u}$ is a solution by first computing the instantiation $\mathbf{e}$ of $\mathbf{E}$ implied by $\mathbf{u}$ (using structural equations) and then checking whether $\mathbf{e}$ is consistent with $\mathbf{e}_1, \mathbf{e}_2$. If the answer is yes, then $Pr(\mathbf{e}_1|\mathbf{u}, \mathbf{e}_2) = 1$, otherwise $Pr(\mathbf{u}, \mathbf{e}_2) = 0$ or $Pr(\mathbf{e}_1|\mathbf{u}, \mathbf{e}_2) = 0$. To show hardness, we show that SAT can be reduced to D-Reverse-MAP under the conditions stated in the theorem. Given a Boolean formula α over variables $\mathbf{U}$, we construct an SCM G_α as in the proof of Theorem 7. SCM G_α has a single leaf node S_α and its root nodes are $\mathbf{U}$. By construction of G_α, we have $Pr(S_\alpha = 1|\mathbf{u}) = 1$ if $\mathbf{u}$ satisfies α and $Pr(S_\alpha = 1|\mathbf{u}) = 0$ otherwise. By choosing $p = 0$, $\mathbf{e}_1 = \{S_\alpha = 1\}$ and $\mathbf{e}_2 = \emptyset$, the D-Reverse-MAP query (is there $\mathbf{u}$ such that $Pr(S_\alpha = 1|\mathbf{u}) > 0$) answers yes iff there is an instantiation $\mathbf{u}$ that satisfies the formula α (SAT query). This concludes our proof.

Appendix F. Review of Elimination Concepts

We review here the standard notions of elimination process, clusters and moral graphs which we use in some of the upcoming proofs; see (Darwiche, 2009, Ch 9) for a detailed treatment.

The *moral graph* of an SCM G is obtained from G by adding an undirected edge between every pair of common parents and then undirecting all edges. *Eliminating* a variable X from a graph G is done by connecting every pair of neighbors for X in G, and then removing node X from G. Eliminating variables from an SCM G is done by eliminating variables from its moral graph G'. Eliminating variables from a moral graph G' using variable order π induces a *graph sequence* $G' =$

10. (Park and Darwiche, 2004b) intended to construct a Bayesian network, but their construction is an SCM since all internal nodes in the network have functional CPTs.

$G_1, \ldots, G_n$ where graph G_{i+1} is obtained by eliminating variable $\pi(i)$ from G_i. We use $G_i(X)$ to denote X and its neighbors in graph G_i. We also use $\mathbf{C}(X)$ to denote the *cluster* of variable X, which is X and its neighbors just before eliminating X. If $X = \pi(i)$, then $\mathbf{C}(X) = G_i(X)$. We also use $\mathbf{C}_i$ to denote the cluster for X, $\mathbf{C}(X)$, in this case.

Appendix G. Proof of Lemma 12

This proof uses the elimination concepts and notations reviewed in Appendix F.

Let G'_m be the moral graph of G' and $\mathbf{C}'_i$ be the cluster induces by eliminating variable X_i from G'. Let n be the number of variables in G. To prove Lemma 12, it suffices to prove the following statement: $\mathbf{C}'_i \subseteq \mathbf{C}_i \cup \{H\}$ for $i = 1, \ldots, n$, which we prove next by induction.

Let $\text{neigh}(X)$ denote the neighbors of X in G_m and let $\text{neigh}'(X)$ denote the neighbors of X in G'_m. Let $\mathbf{Z}$ denote the children of H in G'. For each $Z \in \mathbf{Z}$, let $\mathbf{P}_Z$ denote the parents of Z in G. First, we show the statement holds when $i = 1$. When creating G'_m from G', the introduction of node H would cause two classes of edges that do not exist in G_m to be added to G'_m: 1) (Z, H) for $Z \in \mathbf{Z}$; 2) (Y, H) if Y is a parent of some node $Z \in \mathbf{Z}$, that is Y and H are common parents of some node Z. This means that before the elimination starts, for any node X, if $X \in \mathbf{Z} \cup_{Z \in \mathbf{Z}} P_Z$, we have $\text{neigh}'(X) = \text{neigh}(X) \cup \{H\}$; otherwise $\text{neigh}'(X) = \text{neigh}(X)$. Hence, $C'_1 \subseteq \{C_1\} \cup \{H\}$. Consider now the elimination of X_{i+1} assume that the statement holds for $1, 2, \ldots, i$. We observe that if $\mathbf{C}'_j \subseteq \mathbf{C}_j \cup \{H\}$, then the elimination of X_j would cause only one type of additional edges be added to G'_m, that is (Y, H) for $Y \in \text{neigh}(X_j)$. This is because the elimination of X_j will form a clique among $\text{neigh}(X_j) \cup \{H\}$ in G'_m, but $\text{neigh}(X_j)$ already forms a clique after X_j is eliminated from G_m. This implies that eliminating X_j ($j \leq i$) will never cause any additional edge to be added among any two nodes that are both not H (in other words, all additional edges added are incident on H). Thus, before we eliminate X_{i+1}, we have $\text{neigh}'(X_{i+1}) \subseteq \text{neigh}(X_{i+1}) \cup \{H\}$ and this implies $\mathbf{C}'_{i+1} \cap \mathbf{C}_{i+1} \cup \{H\}$ which concludes the proof.

Appendix H. Proof of Theorem 14

An N-world model is obtained by creating N copies of a directed acyclic graph (DAG) while joining them so a *subset* of their roots are shared (Han et al., 2022). Hence, an objective model as in Definition 2 corresponds to an N-world model except for the addition of mixture node H and its outgoing edges. Before adding the mixture node H, an objective model with n components corresponds to a $3n$-world model so its treewidth is $\leq 3n(w+1) - 1$ by Theorem 13. After adding node H, its treewidth is $\leq 3n(w+1)$ by Lemma 12.

Appendix I. Proof of Theorem 17

This proof uses the elimination concepts and notations reviewed in Appendix F.

For a node X in an SCM G, we use $[X]^k$ to denote its k^{th} duplicate in an n-world model of G. If node X is shared between all n worlds, then $[X]^k = X$ for all k. For a set of variables $\mathbf{X}$, we use $[\mathbf{X}]^k$ to denote $\{[X]^k : X \in \mathbf{X}\}$.

Let G be an SCM, $\mathbf{U}$ be a subset of its roots (unit variables) and let G' be a corresponding objective model with n components. Our proof is based on constructing an augmented objective model G'' by adding edges to G' and then showing that the bounds of Theorem 17 hold for G''. Our proof is based on Lemmas 20 and 22 which we formally state and prove later:

– Lemma 20 complements Theorem 13 by showing that any $\mathbf{U}$-constrained elimination order for an SCM can be converted into a $\mathbf{U}$-constrained elimination order for a corresponding n-world model while preserving the width of the order.

– Lemma 22 concerns the augmentation of an SCM by a root node H and some edges that originate from H. In particular, given a $\mathbf{U}$-constrained elimination order of width w for the SCM, the lemma shows how to construct a $\mathbf{U}$-constrained elimination order for its augmentation with width $\leq \max(w+1, |\mathbf{U}|)$.

We start by showing how to construct the augmented objective model G'' from G'. Let H be the mixture node of G' and $\mathbf{Z} = \{\mathbf{Y}^i, \mathbf{W}^i\}_{i=1}^n$ be the set of all outcome variables in the objective function of Equation (1). We obtain G'' by adding to G' an edge $H \to Z$ for each $Z \in \mathbf{Z}$ if such an edge does not already exist in G'. The edges of G'' are a superset of the edges of G' so it suffices to show that the bounds of Theorem 17 hold for G''. We will next use G^t to denote a triplet (3-world) model of G. We will also use G^b to denote the augmentation of G^t with mixture node H and edges $H \to Z$ for $Z \in \mathbf{Z}$. Note that the augmented objective model G'' corresponds to n copies of G^b that share root nodes $\mathbf{U} \cup \{H\}$. Hence, G'' is an n-world model of G^b.

Let π be a $\mathbf{U}$-constrained elimination order for G with width w. Since G^t is a triplet (3-world) model of G, Theorem 13 tells us that there exists an elimination order π^t of G^t with width w^t such that $w^t \leq 3w+2$ (order π^t will also be $\mathbf{U}$-constrained). Recall that G^b is obtained from G^t by adding a root node H and some edges that emanate from H. By Lemma 22, there exists an elimination order π^b for G^b with width w^b such that $w^b \leq \max(w^t+1, |\mathbf{U}|) = \max(3w+3, |\mathbf{U}|)$. Moreover, if the objective function has a single outcome variable Y, then H has a single child Y in G^b so, also by Lemma 22, we have $w^b \leq 3w+3$. Since G'' is an n-world model of G^b based on roots $\mathbf{U} \cup \{H\}$ of G^b, we have $w'' = w^b$ by Lemma 20. In summary, we have $w' \leq w'' \leq \max(3w+3, |\mathbf{U}|)$. If the objective function has a single outcome variable, we have $w' \leq 3w+3$. This concludes the proof of Theorem 17.

We will next formally state and prove Lemmas 20 and 22 which we used in the above proof.

Lemma 20 *Consider an SCM G, a subset $\mathbf{U}$ of its roots, and a corresponding n-world model G' for G that shares $\mathbf{U}$. If w is the width of a $\mathbf{U}$-constrained elimination order π for G, and w' is the width of the corresponding $\mathbf{U}$-constrained elimination order π' for G', then $w = w'$.*

Given an elimination order π for an SCM G, we can convert it into a corresponding elimination order π' for its n-world model G' (referenced in the above lemma) by replacing each variable $X \notin \mathbf{U}$ in π with its duplicates $[X]^1, [X]^2, \ldots, [X]^n$, as in Definition 2 in (Han et al., 2022). If π is $\mathbf{U}$-constrained, then π' will also be $\mathbf{U}$-constrained. Moreover, we define a graph sequence for the n-world model $G'_1, G'_2, \ldots, G'_n$ where G'_1 is the moral graph of G', and G'_{i+1} is obtained by eliminating all duplicates of variable $\pi(i)$, i.e. $[\pi(i)]^1, [\pi(i)]^2, \ldots, [\pi(i)]^n$, from G'_i.

Proof Suppose we eliminate variables from G/G' using orders π/π'. We claim that at every elimination step i, the following properties hold:

A) For each node $X \notin \mathbf{U}$, $G'_i([X]^k) = [G_i(X)]^k$

B) For each node $U \in \mathbf{U}$, $G'_i(U) = \bigcup_{k=1}^n [G_i(U)]^k$

We next show that properties A), B) imply $w' = w$ and then prove these properties. Let $Y = \pi(i)$. If $Y \notin \mathbf{U}$, then when its duplicate $[Y]^k$ is eliminated from G', we have $\mathbf{C}'([Y]^k) = [\mathbf{C}(Y)]^k$. If $Y \in \mathbf{U}$, then when Y is eliminated from G', we have $\mathbf{C}'(Y) = \bigcup_{k=1}^{n} [G_i(Y)]^k = G_i(Y) = \mathbf{C}(Y)$ since all non-shared nodes have been eliminated before Y, i.e. $[G_i(Y)]^k = G_i(Y)$. This means that the cluster induced by eliminating a variable from G' always has the same size as the cluster induced by eliminating the corresponding variable from G, which implies $w' = w$.

We next prove properties A), B) by induction. By definition of an n-world model, these properties hold initially for G'_1. Suppose they hold for G'_i and consider G'_{i+1}. Let $Y = \pi(i)$. Then G'_{i+1} is the result of eliminating nodes $[Y]^1, \ldots, [Y]^n$ from G'_i. We consider two cases.

Case: $Y \notin \mathbf{U}$. Consider each node Z in G'_i. If Z is not a neighbor of $[Y]^1, \ldots, [Y]^n$ in G'_i, then $G'_{i+1}(Z)$ will not be affected by the elimination of $[Y]^1, \ldots, [Y]^n$ and the properties hold by the induction hypothesis. Otherwise, node Z falls into two cases: A) a duplicate $[X]^k$ of a node $X \notin \mathbf{U}$ B) a shared node $U \in \mathbf{U}$.

A) by the induction hypothesis, neighbors of $[X]^k$ in G'_i must belong to the k-th world, so $[X]^k$ can only be a neighbor of the k-th duplicate $[Y]^k$. This means that $G'_{i+1}([X]^k)$ can only be affected by the elimination of $[Y]^k$. By definition of variable elimination, we have:

$$\begin{aligned}
G'_{i+1}([X]^k) &= G'_i([X]^k) \cup G'_i([Y]^k) \setminus \{[Y]^k\} \\
&= [G_i(X)]^k \cup [G_i(Y)]^k \setminus \{[Y]^k\} && \text{by the induction hypothesis} \\
&= [G_i(X) \cup G_i(Y) \setminus \{Y\}]^k \\
&= [G_{i+1}(X)]^k && \text{by definition of variable elimination}
\end{aligned}$$

This proves property A).

B) by the induction hypothesis, U must be a neighbor of all duplicates $[Y]^1, \ldots, [Y]^n$. We have:

$$\begin{aligned}
G'_{i+1}(U) &= G'_i(U) \bigcup_{k=1}^{n} G'_i([Y]^k) \setminus \{[Y]^k\}_{k=1}^{n} \\
&= \big(\bigcup_{k=1}^{n} [G_i(U)]^k\big)\big(\bigcup_{k=1}^{n} [G_i(Y)]^k\big) \setminus \{[Y]^k\}_{k=1}^{n} && \text{by the induction hypothesis} \\
&= \bigcup_{k=1}^{n} [G_i(U)]^k \cup [G_i(Y)]^k \setminus \{[Y]^k\} \\
&= \bigcup_{k=1}^{n} [G_i(U) \cup G_i(Y) \setminus \{Y\}]^k \\
&= \bigcup_{k=1}^{n} [G_{i+1}(U)]^k && \text{by definition of variable elimination}
\end{aligned}$$

This proves property B).

Case: $Y \in \mathbf{U}$. In this case, G'_i only contains nodes in $\mathbf{U}$. Property A) holds trivially. And the relation in property B) reduces to $G'_i(U) = \bigcup_{k=1}^{n} [G_i(U)]^k = G_i(U)$. By the induction hypothesis, we know $G'_i = G_i$ and thus $G'_{i+1} = G_{i+1}$. Property B) holds. This concludes the proof. ■

The proof of Lemma 22 requires the following result on eliminating variables from graphs.

Lemma 21 *Consider a DAG G, a subset $\mathbf{U}$ of its nodes, and a node H in G where $H \notin \mathbf{U}$. Let G_1 be the moral gragh of G, and G_2 be the result of eliminating all nodes other than $\{H\} \cup \mathbf{U}$ from G_1. For any node $X \in \mathbf{U}$, X is adjacent to H in G_2 if and only if there exists a path between X and H in G_1 that does not include a node in $\mathbf{U} \setminus \{X\}$.*

Proof We first prove the if direction. Suppose there exists such a path $(X, \ldots, Z_1, Y, Z_2, \ldots, H)$ in G_1. Eliminating node Y from G_1 will lead to a path $(X, \ldots, Z_1, Z_2, \ldots, H)$. Since nodes in $\mathbf{U} \setminus \{X\}$ cannot appear along this path, eliminating all nodes other than $\{H\} \cup \mathbf{U}$ will lead to the edge (X, H) in G_2. We next prove the only-if direction by contraposition. Suppose there is no path between X and H in G_1 that does not include a node in $\mathbf{U} \setminus \{X\}$. There are two cases: 1) there is no path between X and H; 2) every path between X and H includes at least one node $U \in U \setminus \{X\}$, which has the form $(X, \ldots, U, \ldots, H)$. In the first case, X and H will be disconnected in G_2. In the second case, eliminating all nodes other than $\{H\} \cup \mathbf{U}$ from such paths will lead to $X \to U \to H$, so X cannot be directly adjacent to H in G_2. This concludes the proof. ■

Lemma 22 *Consider an SCM G and a subset $\mathbf{U}$ of its roots. Suppose SCM G' is obtained from G by adding a root node H as a parent of some nodes $\mathbf{Z}$ in G where $\mathbf{Z} \cap \mathbf{U} = \emptyset$. Let π be a $\mathbf{U}$-constrained elimination order for G, and let π' be a $\mathbf{U}$-constrained elimination order of G' obtained from π by placing H just before variables $\mathbf{U}$. If π has width w and π' has width w', then $w' \leq \max(w + 1, |\mathbf{U}|)$. Moreover, if H has a single child in G', then $w' = w + 1$.*

Proof Let $\mathbf{X}$ denote variables other than $\mathbf{U}$ in G, and let $\mathbf{U}' = \mathbf{U} \cup \{H\}$. Suppose we first eliminate variables $\mathbf{X}$, then H, and finally $\mathbf{U}$ from G' using order π'. This results in a graph sequence $G'_1, \ldots, G'_j, G'_{j+1/2}, G'_{j+1}, \ldots, G'_{j+k}$ where $j = |\mathbf{X}|$ and $k = |\mathbf{U}|$. Here, $G'_{j+1/2}$ is obtained by eliminating all variables $\mathbf{X}$ from G'_1, and G'_{j+1} is obtained by eliminating H from $G'_{j+1/2}$. We claim:

- if $i \leq j$, then for each node $X \neq H$ in G'_i, we have $G'_i(X) \subseteq G_i(X) \cup \{H\}$.
- if $i > j$, then for each node $X \neq H$ in G'_i, we have $G'_i(X) \subseteq \mathbf{U}'$. Moreover, if H has a single child in G', then $G'_i(X) = G_i(X)$.

We first show that the above claim implies the lemma, and then follow by proving the claim. Suppose we are eliminating variable Y from G'. If $Y \notin \mathbf{U}'$ then $i \leq j$ and the above claim implies $\mathbf{C}'(Y) \subseteq \mathbf{C}(Y) \cup \{H\}$. If $Y \in \mathbf{U}'$ then $i > j$ and the above claim implies $\mathbf{C}'(Y) \subseteq \mathbf{U}'$, and $\mathbf{C}'(Y) = \mathbf{C}(Y)$ when H has a single child. This guarantees the statement of the lemma: $w' \leq \max(w + 1, |\mathbf{U}|)$, and $w' = w + 1$ if H has a single child in G'.

We next prove our claim by induction. Let $\mathbf{Z}$ denote the children of H in G'. When constructing the moral graph G'_1 from G', the introduction of node H causes two classes of edges that do not exist in G_1 to be added to G'_1: (Z, H) for $Z \in \mathbf{Z}$, and (Y, H) if Y is a parent of some node $Z \in \mathbf{Z}$, that is Y and H are common parents of some node Z. All of these extra edges are incident on H, meaning that for any node X in G'_1, $G'_1(X) \subseteq G_1(X) \cup \{H\}$. Thus, our claim holds for G'_1. Next, assume our claim holds for G'_i (induction hypothesis) and consider G'_{i+1}. We have two cases.

Case: $i \leq j$. Let $Y = \pi(i)$. Consider each node X in G'_{i+1}. If node X is not a neighbor of Y in G_i/G'_i, then X is not affected by the elimination of Y, i.e., $G'_{i+1}(X) = G'_i(X)$ and $G_{i+1}(X) = G_i(X)$. So the claim holds by the induction hypothesis. Otherwise, we can bound $G'_{i+1}(X)$ as follows:

$$\begin{aligned}
G'_{i+1}(X) &= G'_i(X) \cup G'_i(Y) \setminus \{Y\} && \text{by the definition of elimination} \\
&\subseteq \big(G_i(X) \cup \{H\}\big) \cup \big(G_i(Y) \cup \{H\}\big) \setminus \{Y\} && \text{by the induction hypothesis} \\
&\subseteq \big(G_i(X) \cup G_i(Y) \setminus \{Y\}\big) \cup \{H\} \\
&\subseteq G_{i+1}(X) \cup \{H\}
\end{aligned}$$

Case: $i > j$. For this case, G'_i only contains nodes in $\mathbf{U}'$. It is trivial that $G'_i(X) \subseteq \mathbf{U}'$ for each node X in G'_i. Recall that eliminating H from $G'_{j+1/2}$ results in G'_{j+1}. By the induction hypothesis, all extra edges in $G'_{j+1/2}$ that do not exist in G_{j+1} must be incident on H. Consider the special case where H has a single child in G'. We claim that in this case, every two nodes in $G'_{j+1/2}(H)$ are adjacent in $G'_{j+1/2}$, meaning that the neighbors of H already forms a clique in $G'_{j+1/2}$. Thus, eliminating H from $G'_{j+1/2}$ will not add any fill-in edges in G'_{j+1}. This guarantees $G'_{j+1} = G_{j+1}$, i.e, $G'_i(X) = G_i(X)$ for all $i >= j+1$. We finally turn to proving this claim by contradiction. Suppose that node U_1 and U_2 are neighbors of H in $G'_{j+1/2}$ but are not adjacent in $G'_{j+1/2}$. By Lemma 21, in G'_1, there must be a path P_1 between U_1 and H that does not include nodes in $\mathbf{U} \setminus \{U_1\}$, and a path P_2 between U_2 that does not include nodes in $\mathbf{U} \setminus \{U_2\}$. Since H is a root and only has one child Z in G', P_1 must have the form $(U_1, \ldots, Z, H)$ in G'_1 and P_2 must have the form $(U_2, \ldots, Z, H)$ in G'_1. Thus, there must be a path $(U_1, \ldots, Z, \ldots, U_2)$ in G'_1 that does not contain nodes in $\mathbf{U}' \setminus \{U_1, U_2\}$. By Lemma 21, after eliminating all nodes other than U' from G'_1, U_1 and U_2 must be adjacent in G'_{j+1}. This leads to a contradiction. ■

Appendix J. Preliminary Experiment

We provide next a preliminary experiment in which we compare the complexities of three algorithms: (1) MAP_VE (Algorithm 1) for computing MAP (operates on an SCM); (2) RMAP_VE (Algorithm 3) for solving unit selection (operates on an objective model); and (3) a baseline, brute-force method for solving unit selection (operates on a twin-model). We consider the complexities of these algorithms on random SCMs generated using the method in (Han et al., 2022). This method generates a random DAG and then ensures that each internal node in the DAG has at least one parent which is a root node by adding additional root nodes (to mimic the structure of SCMs). Such DAGs tend to have many root nodes and are particularly difficult for algorithms whose complexity is exponential in the constrained treewidth, like MAP_VE and RMAP_VE, as we show later.

Given a random SCM structure, we randomly select different percentages of roots to be unit variables $\mathbf{U}$. We assume the objective function of (Li and Pearl, 2019) given in Equation (2). This function has a single outcome variable which we choose randomly from the SCM leaves. Moreover, as discussed earlier, this function requires only a twin model when constructing the objective model since it does not include evidence variables. We do not prune the SCMs used by MAP_VE, the objective models used by RMAP_VE, or the twin models used by the bruteforce method (see (Darwiche, 2009, Ch. 6)) so the choice of interventional variables do not affect our complexity analysis (no evidence variables in the objective function of Equation (2)). The time

complexity of MAP_VE is $O(n \cdot \exp(w))$, where n is the number of SCM nodes and w is the width of a $\mathbf{U}$-constrained elimination order for the SCM. The time complexity of RMAP_VE is $O(n_1 \cdot \exp(w_1))$, where n_1 is the number of nodes in the objective model and w_1 is the width of a $\mathbf{U}$-constrained elimination order for the objective model. The bruteforce method enumerates every instantiation $\mathbf{u}$ and returns the one maximizing the objective $L(\mathbf{u})$. Its time complexity is $O(n_2 \cdot \exp(w_2))$, where n_2 is the number of nodes in the twin model used to evaluate $L(\mathbf{u})$ and $w_2 = |\mathbf{U}|+$the width of an *unconstrained* elimination order for the twin model. Hence, we compare the complexities of these three algorithms by reporting the number of nodes n, n_1, n_2 and the corresponding widths w, w_1, w_2. These are depicted in Table 1 which also reports the number of SCM roots (R) and the percentage of roots used as unit variables (ur).

Before we highlight the outcomes of this experiment, we provide some insights into the class of used SCMs and their difficulty. We next characterize a class of problems for which the $\mathbf{U}$-constrained treewidth is no smaller than the number of unit variables, $|\mathbf{U}|$. The random SCMs we use in this experiment resemble this class of problems given how they are constructed.

Definition 23 *Consider a connected DAG G and a subset $\mathbf{U}$ of its roots. We say that $\mathbf{U}$ are external to G if the DAG remains connected after removing nodes $\mathbf{U}$ and all their incident edges.*

Markovian SCMs (each root node has a single child) satisfy the above condition.

Lemma 24 *Consider a connected DAG G, a subset $\mathbf{U}$ of its roots, and its moral graph G'. Let $\mathbf{S}$ be the subset of $\mathbf{U}$ such that for every two nodes U_1 and U_2 in $\mathbf{S}$, there exists a path between U_1 and U_2 in G' that does not include any node in $\mathbf{U} \setminus \{U_1, U_2\}$. If π is a $\mathbf{U}$-constrained elimination order of G that has width w, then we have $w \geq |\mathbf{S}|$.*

Proof By Lemma 21, every two nodes U_1 and U_2 in $\mathbf{S}$ will be adjacent after all nodes other than $\mathbf{U}$ are eliminated from G'. Thus, nodes in $\mathbf{S}$ will form a clique after all nodes other than $\mathbf{U}$ are eliminated. This leads to a cluster of size $|\mathbf{S}|$ during the elimination process, so $|\mathbf{S}|$ is a lower bound for the width of any $\mathbf{U}$-constrained elimination order. ■

Our main insight is stated in the following corollary which shows that MAP_VE and RMAP_VE must be exponential in the number of unit variables for the class of SCMs (and unit variables) identified by Definition 23. The baseline method can be significantly worse since it is exponential in the number of unit variables plus the unconstrained treewidth of the twin model.

Corollary 25 *Consider a connected SCM G, a subset $\mathbf{U}$ of its roots, and a $\mathbf{U}$-constrained elimination order π with width w. If $\mathbf{U}$ are external to G, then $w \geq |\mathbf{U}|$.*

Proof Consider any two nodes U_1 and U_2 in $\mathbf{U}$. Suppose that X_1 is a child of U_1 and X_2 is a child of U_2. Since $\mathbf{U}$ are external to G, there exists a path $(X_1, \ldots, X_2)$ in the moral graph of G that does not include any nodes in $\mathbf{U}$. Thus, there exists a path $(U_1, X_1, \ldots, X_2, U_2)$ that do not include any nodes in $\mathbf{U} \setminus \{U_1, U_2\}$. By Lemma 24, this implies $w \geq |\mathbf{U}|$ since $\mathbf{S} = \mathbf{U}$. ■

We can now highlight the patterns in Table 1. The complexities of MAP_VE and RMAP_VE are relatively close with the latter being more expensive than the former. Moreover, the gap between them narrows as the number of SCM variables (n) and the number of unit variables (ur) increase.

ur			20%				40%				60%			
n	n_2	R	n_1	w	w_1	w_2	n_1	w	w_1	w_2	n_1	w	w_1	w_2
10	14	6	52	5.5	7.2	7.3	49	5.5	7.4	8.3	46	5.5	7.5	9.3
15	21	9	82	7.4	10.0	10.6	76	7.4	10.2	12.6	70	7.5	11.0	14.6
20	30	12	116	10.1	14.0	16.0	110	10.1	14.5	18.0	101	10.1	15.4	21.0
25	37	15	140	11.0	16.8	19.7	131	11.0	17.4	22.7	122	11.2	18.4	25.7
30	43	17	163	11.3	18.8	21.6	154	11.4	19.2	24.6	142	11.8	20.7	28.6
35	50	19	190	12.4	21.4	24.2	178	12.4	21.9	28.2	166	12.8	23.5	32.2
40	57	22	218	13.3	24.3	27.8	206	13.6	24.9	31.8	191	14.3	26.6	36.8
45	64	24	246	14.3	26.6	29.8	231	14.4	27.2	34.8	216	15.6	28.9	39.8

ur			80%				100%			
n	n_2	R	n_1	w	w_1	w_2	n_1	w	w_1	w_2
10	14	6	43	5.5	7.7	10.3	37	6.3	8.3	12.3
15	21	9	64	8.2	11.8	16.6	58	9.6	12.6	18.6
20	30	12	95	10.5	15.6	23.0	86	12.8	16.4	26.0
25	37	15	113	12.8	18.8	28.7	104	15.9	19.9	31.7
30	43	17	133	13.8	21.2	31.6	121	18.0	21.6	35.6
35	50	19	154	15.6	24.0	36.2	142	19.6	23.6	40.2
40	57	22	179	17.6	27.0	40.8	164	23.0	26.6	45.8
45	64	24	201	20.1	30.6	44.8	186	25.6	29.2	49.8

Table 1: Comparing the complexities of MAP_VE for solving MAP ($n \exp(w)$), RMAP_VE for solving unit selection ($n_1 \exp(w_1)$), and the bruteforce method for solving unit selection ($n_2 \exp(w_2)$). Each data point is an average over 25 runs. All elimination orders are computed using the minfill heuristic (Kjærulff, 1990).

Note that according to Theorem 17, $w_1/w \leq 2$ yet Table 1 shows that this ratio can be significantly smaller than 2. Finally, the bruteforce method is significantly worse than RMAP_VE and the gap between the two grows as the number of SCM variables (n) and unit variables (ur) increase.

We close this discussion by identifying a class of problems with an unbounded number of unit variables $\mathbf{U}$ yet a bounded $\mathbf{U}$-constrained treewidth. This class is depicted in Figure 5. The $\mathbf{U}$-constrained treewidth is 3, which can be shown using the $\mathbf{U}$-constrained elimination order $S_1, \ldots, S_n, \ldots, U_1, \ldots, U_n$. This is a class of problems for which unit selection using RMAP_VE is tractable even when the number of unit variables is unbounded, assuming one uses a suitable objective function (e.g., the benefit function of (Li and Pearl, 2019) given in Equation (2)).

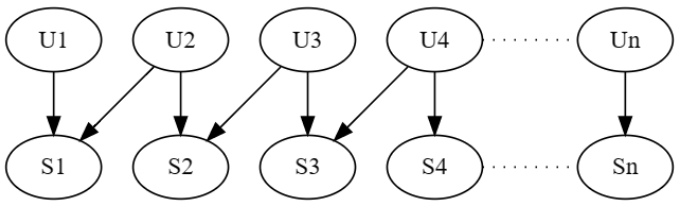


Figure 5: The unit variables are $\mathbf{U} = U_1, \ldots, U_n$. The $\mathbf{U}$-constrained treewidth is 3.