A Fully Polynomial Time Approximation Scheme for Updating Credal Networks of Bounded Treewidth and Number of Variable States

Denis D. Mauá
IDSIA, Switzerland
denis@idsia.ch

Cassio P. de Campos
IDSIA, Switzerland
cassio@idsia.ch

Marco Zaffalon
IDSIA, Switzerland
zaffalon@idsia.ch

Abstract

Credal networks lift the precise probability assumption of Bayesian networks, enabling a richer representation of uncertainty in the form of closed convex sets of probability measures. The increase in expressiveness comes at the expense of higher computational costs. In this paper we present a new algorithm which is an extension of the well-known variable elimination algorithm for computing posterior inferences in extensively specified credal networks. The algorithm efficiency is empirically shown to outperform a state-of-the-art algorithm. We then provide the first fully polynomial time approximation scheme for inference in credal networks with bounded treewidth and number of states per variable.

Keywords. Probabilistic graphical models, credal networks, approximation scheme, valuation algebra.

1 Introduction

Credal networks [11] are generalizations of Bayesian networks that allow for a richer representation of uncertainty in the form of set-valued probabilities—in contrast to the sharp numeric values required by their Bayesian counterpart. They are models of imprecise probability as advocated by Walley [18]. In a nutshell, credal networks rely on a directed acyclic graph (DAG) to encode a compact and computationally efficient representation of a closed convex set of joint probability mass functions over a set of variables, much in the same way that Bayesian networks do for single joint probability mass functions. Namely, credal networks respect the local Markov condition that each variable (uniquely represented by a node in the DAG) is (strongly) independent of its non-descendant non-parents conditional on its parents. Strong independence is justified by a sensitivity analysis interpretation, where we assume that there exists a single probability mass function representing our knowledge which we cannot know precisely for lack of resources; epistemic irrelevance, on the other hand, is arguably more consistent with a behavioral interpretation of inherent imprecision [18]. In the following, we assume credal networks to operate under strong independence.

In order to enable efficient computation, additional constraints need to be imposed to the set-valued specifications of the local probabilities. The two most common choices are *extensively specified* sets, in which local models are given as sets of probability potentials, and *separately specified* sets, in which local models are specified as collections containing one set of probability mass functions for each configuration of the parents. Separately specified networks can be mapped to extensively specified and vice-versa [2].

There is also another subtlety when computing with such local models, which concerns the way they are represented in a computer. The sets of local (conditional) probability mass functions can be encoded either as sets of points (e.g., the sets of vertices of a convex polytope), or as sets of (linear) inequalities. Although these two encodings can represent any finitely-generated closed convex set, moving from an inequality-based encoding to a vertex-based encoding can dramatically increase the length of the representation of the local models. For example, a simple 8-dimensional polytope specified by 729 inequalities has between 5 thousand and 12 billion vertices [4].

Inference with credal networks has been theoretically and empirically shown to be a difficult problem. For example, computing exact marginals in credal networks is known to be NP-hard even for polytree-shaped networks, a particular case that can be computed in polynomial time in Bayesian networks [7]. Despite the hardness of the problem, several algorithms are known to perform reasonably well under certain conditions. Most notably, the 2U algorithm [12], which computes exact posterior bounds in polytree-shaped credal networks with binary variables, continues to be the only known polynomial time algorithm available, and its generalizations to arbitrary networks (e.g., the GL2U [3]), which perform approximate inference, are among the fastest algorithms. A notable example, against which we compare our results in this paper, is the algorithm of de Campos and Cozman [8], which

algorithm	complexity	topology	inference	representation
2U [12]	polynomial	polytree	exact	inequality
GL2U [3]	polynomial	all	approximate	vertex
A/R+ [16]	exponential	polytree	approximate	inequality
IP [8]	exponential	all	exact/approx.	inequality
ML [6]	exponential	all	exact/approx.	inequality
HC [9]	exponential	all	exact/approx.	vertex

Table 1: Comparison of some existing algorithms for inference in credal networks.

finds exact posterior bounds in general networks by converting the problem into a mixed integer program, which can be solved exactly for small networks, or relaxed to provide approximate results in large networks. Other approaches mix branch-and-bound methods for exact inference and local searches for approximate results [6, 9, 16]. Table 1 contrasts some of the available algorithms. To date, no algorithm is known to provide approximations within given bounds in polynomial time. Recently, de Cooman et al. [10] developed a polynomial time algorithm for tree-shaped credal networks, but it operates under epistemic irrelevance.

In this paper, we present a new algorithm for computing exact posterior bounds in extensively specified credal networks encoded by vertices, as well as a fully polynomial time approximation scheme (FPTAS) for networks with bounded treewidth and number of states per variable. We begin by stating the basic elements of our formalism (Section 2), followed by a formal definition of inference in extensively specified credal networks (Section 3). Then we present a modified variable elimination algorithm for exact inference, which has worst-case complexity exponential in both the treewidth of the graph and the size of local sets (Section 4). We address this issue by devising an FPTAS (Section 5). Experiments showing the performance of the algorithms are presented and discussed in Section 6. Finally, Section 7 contains our concluding thoughts.

Due to the limited space, we only present proofs for the most important results.

2 An Algebra of Ordered Potentials

In this section, we introduce the main ingredients of the message passing algorithms that we present later as well as the basic results needed to guarantee the correctness and efficiency of computations.

From an algebraic viewpoint, the primitive entities of our formalism are the so-called *labeled valuations* (ϕ, x), which encode information about a (local) domain through a *valuation* ϕ and a set of *variables* x. Here we adopt the equivalent notation ϕ_x to denote the pair (ϕ, x). More concretely, valuations can take as straightforward forms as bounded real-valued functions (Section 2.2), or represent more complicated objects such as sets of pairs of probability potentials (Section 2.3).

The set of all variables we consider relevant to a problem, denoted by $\mathcal{U}$, is the largest set of variables that can be considered for a (labeled) valuation in our setting, which we assume to be bounded. We write variables with capital letters (e.g., $X_1, \ldots, X_n \in \mathcal{U}$) and sets of variables in lower case (e.g., $x = \{X_1, \ldots, X_n\}$). Any variable X is assumed to be associated with a finite set of values Ω_X called its *frame*. The elements of Ω_X are called states. If x is a set of variables, the domain Ω_x is given by the Cartesian product of the frames of variables in x, $\Omega_x \triangleq \times_{X \in x} \Omega_X$. Any element of Ω_x is called a configuration. If $\mathbf{x}$ is a configuration in Ω_x, the notation $\mathbf{x}^{\downarrow y}$ denotes the projection of $\mathbf{x}$ onto $y \subseteq x$, with $\mathbf{x}^{\downarrow \emptyset} \triangleq \lambda$, where λ denotes the null element that does not appear in any frame.

The set of all valuations (ϕ, x) over a subset $x \subseteq \mathcal{U}$ is denoted by Φ_x. The set of all valuations is denoted by $\Phi \triangleq \bigcup_{x \subseteq \mathcal{U}} \Phi_x$. The algebra comes with two basic operations of *combination* and *marginalization*. Intuitively, combination represents aggregation of two pieces of information. If ϕ_x and ϕ_y are two arbitrary valuations, then $\phi_x \times \phi_y$ is a valuation $\phi_{x \cup y}$ with domain $\Omega_{x \cup y}$. Marginalization, on the other hand, acts by coarsening information. If ϕ_x is a valuation then the marginal $\phi_x^{\downarrow y}$ is a valuation with domain Ω_y. Sometimes, it is convenient to define the elimination operation, which is in a one-to-one correspondence to marginalization. Formally, if ϕ_x is a valuation then $\phi_x^{-y} \triangleq \phi_x^{\downarrow x \setminus y}$ is the result of the elimination of variables in y. When clear from the context, we write Y to denote a singleton $y = \{Y\}$, for example $\phi_x^{-Y} = \phi_x^{\downarrow x \setminus \{Y\}}$. A system $(\Phi, \mathcal{U}, \times, \downarrow)$ closed under combination and marginalization is said to be a *valuation algebra* if it satisfies the following three axioms [15, 17].

(A1) Combination is commutative and associative.

(A2) For $y \subseteq x \subseteq z$, $\left(\phi_z^{\downarrow x}\right)^{\downarrow y} = \phi_z^{\downarrow y}$.

(A3) If $x \subseteq z \subseteq x \cup y$ then $(\phi_x \times \phi_y)^{\downarrow z} = \phi_x \times \phi_y^{\downarrow z \cap y}$.

The purpose of a valuation algebra is the computation of marginals of the form $\left(\times_i \phi_{u_i}\right)^{\downarrow y}$, where the joint valuation $\bigtimes_i \phi_{u_i}$ is computationally too expensive to be obtained explicitly. The complexity of the operations of combination and marginalization is given by the size of the valuations involved, which is in general a function of the cardinality of the domain. Hence, as a rule-of-thumb, the larger the domain of a valuation the more expensive are the operations involving it. The axioms of valuation algebras provide the necessary framework for breaking down the computation of costly marginals into a sequence of computations of marginals over smaller domains. The pseudocode in Algorithm 1 exhibits the variable elimination procedure (also known as fusion algorithm), which more efficiently computes marginals of factorized valuations.

Algorithm 1: Variable Elimination

```
input  : A finite set of valuations $\Psi$, a set of target
         variables $y \subset \mathcal{U} \triangleq \bigcup_{\phi_u \in \Psi} u$, and an ordering
         $o = (X_1, \ldots, X_n)$ of the variables in $\mathcal{U} \setminus y$
output: The marginal $(\bigtimes_{\phi \in \Psi} \phi)^{\downarrow y}$

for $i \leftarrow 1$ to $n$ do
    Set $\mathcal{B}_i \leftarrow \{\phi_u \in \Psi : X_i \in u\}$ ;
    Compute $\Psi^i \triangleq (\bigtimes_{\phi \in \mathcal{B}_i} \phi)^{-X_i}$ ;
    Set $\Psi \leftarrow (\Psi \setminus \mathcal{B}_i) \cup \{\Psi^i\}$;
end
return $\Gamma \triangleq \bigtimes_{\phi \in \Psi} \phi$;
```

Instead of computing a valuation $\bigtimes_{\phi \in \Psi} \phi$ over a large domain $\Omega_{\mathcal{U}}$ and then marginalizing to y, the algorithm computes marginals $(\bigtimes_{\phi \in \mathcal{B}_i} \phi)^{-X_i}$ over possibly much smaller domains. The overall complexity of the algorithm is given by the size of the largest valuation Ψ^i generated at the loop step. If such a size is bounded then (A1)–(A3) are sufficient to show that the algorithm efficiently outputs the desired marginal [15].

Some optimization tasks like the credal network inferences we aim at here admit a partial ordering over the valuations. Let $\leq$ denote a partial order over Φ (i.e., a reflexive, antisymmetric and transitive relation). An *ordered valuation algebra* [13] is a system $(\Phi, \mathcal{U}, \times, \downarrow, \leq)$, where $(\Phi, \mathcal{U}, \times, \downarrow)$ is a valuation algebra and $\leq$ is monotonic with respect to $\times$ and $\downarrow$:

(A4) If $\phi_x \leq \psi_x$ and $\phi_y \leq \psi_y$ then $(\phi_y \times \phi_x) \leq (\psi_y \times \psi_x)$ and $\phi_x^{\downarrow y} \leq \psi_x^{\downarrow y}$.

Given a finite set of ordered valuations $\Psi \subseteq \Phi$, we say that $\phi \in \Psi$ is *maximal* if for all $\psi \in \Psi$ such that $\phi \leq \psi$ it holds that $\psi \leq \phi$. The operation $\max(\Psi)$ returns the set of maximal valuations of a set Ψ. Given any relation R on Ψ, a subset $\Psi' \subseteq \Psi$ is called an *R-covering* of Ψ if for every $\phi \in \Psi$ there is $\psi \in \Psi'$ such that $\phi R \psi$. For example, the set $\max(\Psi)$ is a $\leq$-covering for Ψ.

2.1 Set-Valuations

The algorithms we develop use the more complex entities of sets of valuations, called *set-valuations*. Theses entities can nevertheless be casted in the algebra of valuations, and manipulated by the variable elimination algorithm to produce sets of marginal valuations.

Let 2^{Φ_x} denote the power set of Φ_x, that is, the set of all subsets of it. Thus, 2^{Φ} denotes the set of all subsets of valuations in Φ. If $\Psi_x \in 2^{\Phi_x}$ and $\Psi_y \in 2^{\Phi_y}$, we define their set-combination $\otimes$ as the set-valuation resulting from element-wise combination of their elements, $\Psi_x \otimes \Psi_y \triangleq \{\phi_x \times \phi_y : \phi_x \in \Psi_x, \phi_y \in \Phi_y\}$. Likewise, we define the set-marginalization operation $\Downarrow$ on 2^{Φ} as the element-wise marginalization of the valuations in a set, $\Psi_x^{\Downarrow y} \triangleq \{\phi_x^{\downarrow y} : \phi_x \in \Psi_x\}$.

Proposition 1. *The system* $(2^{\Phi}, \mathcal{U}, \otimes, \Downarrow)$ *of set-valuations with set-combination and set-marginalization is a valuation algebra.*

The exact variable elimination algorithm we develop in Section 4 obtains its (relative) efficiency by propagating only maximal valuations. Let $\max(2^{\Phi}) \triangleq \{\max(\Psi) : \Psi \in 2^{\Phi}\}$ denote the set of all sets of maximal valuations in 2^{Φ}. We define the max-combination $\oplus$ and max-marginalization $\downdownarrows$ as $\Psi_x \oplus \Psi_y \triangleq \max(\Psi_x \otimes \Psi_y)$ and $\Psi_x^{\downdownarrows y} \triangleq \max(\Phi_x^{\Downarrow y})$.

Proposition 2. *The system* $(\max(2^{\Phi}), \mathcal{U}, \oplus, \downdownarrows)$ *of maximal set valuations with max-combination and max-marginalization is also a valuation algebra.*

If $(\Phi_1, \mathcal{U}, \times_1, \downarrow_1)$ and $(\Phi_2, \mathcal{U}, \times_2, \downarrow_2)$ are two valuation algebras, we say that a mapping $h : \Phi_1 \rightarrow \Phi_2$ is a *homomorphism* if for any $\phi_x, \phi_y \in \Phi_1$ we have that $h(\phi_x) \times_2 h(\phi_y) = h(\phi_x \times_1 \phi_y)$ and $h(\phi_x)^{\downarrow_2 y} = h(\phi_x^{\downarrow_1 y})$. Thus, if we are interested in computing $h(\phi_1^{\downarrow_1 y})$ for some valuation $\phi_1 \in \Phi_1$ that we know that factorizes as $\phi_1 = \psi_1 \times_1 \cdots \times_1 \psi_m$, we can equivalently obtain $(h(\psi_1) \times_2 \cdots \times_2 h(\psi_m))^{\downarrow_2 y}$, which might be computationally more convenient. The following result relates the algebras of set-valuations and maximal set-valuations.

Proposition 3. $\max$ *is a homomorphism from* $(2^{\Phi}, \mathcal{U}, \otimes, \Downarrow)$ *to* $(\max(2^{\Phi}), \mathcal{U}, \oplus, \downdownarrows)$.

Since the set of maximal elements of a set is in the worst case as large as the set itself, but often much smaller, the homomorphism $\max$ allows us to conveniently obtain a set of maximal marginals $\max([\bigotimes_i \Psi_{x_i}]^{\Downarrow y})$ by computing the equivalent $[\bigoplus_i \max(\Psi_{x_i})]^{\downdownarrows y}$. Recall that $\otimes$ is defined as element-wise combination of valuations in the cartesian product, and assume that the set-valuations Ψ_{x_i} can not be factorized as combinations of other set-valuations. Hence, the set $\bigotimes_i \Psi_{x_i}$ is exponentially large in the size of each Ψ_{x_i} and often intractable. On the other hand, the combination of maximal set-valuations $\bigoplus_i \max(\Psi_{x_i})$ can mitigate the exponential explosion if the number of maximal points is kept bounded after each pairwise combination. For instance, if each of the local maximal sets $\max(\Psi_{x_i})$ is half as large as its original set Ψ_{x_i}, then computing $\max([\bigotimes_i \max(\Psi_{x_i})]^{\Downarrow y})$ involves $O(2^n)$ less computations than $\max([\bigotimes_i \Psi_{x_i}]^{\Downarrow y})$. The speed up strongly depends on the number of non-maximal elements that are discarded after each max-combination.

In the rest of this section we introduce the concrete valuation algebras our framework relies on.

2.2 Probability Potentials

Probability potentials are perhaps the most common example of valuation algebras. They generalize (conditional) probability mass functions. If $x \subseteq \mathcal{U}$ is a nonempty set of variables, we define a *potential* p_x as a mapping from Ω_x to the set of nonnegative reals. A potential $p_\emptyset$ over the empty set is defined as a nonnegative real number. The size of a potential p_x is the cardinality of its domain. The following operations are defined over potentials. Combination of potentials is done by element-wise multiplication: for $\mathbf{z} \in \Omega_{x \cup y}$,

$$(p_x \times p_y)(\mathbf{z}) \triangleq p_x(\mathbf{z}^{\downarrow x}) p_y(\mathbf{z}^{\downarrow y}) \,. \tag{1}$$

Marginalization is defined as the sum of compatible elements. For $\mathbf{y} \in \Omega_y$,

$$p_x^{\downarrow y}(\mathbf{y}) \triangleq \sum_{\mathbf{x} \in \Omega_x : \mathbf{x}^{\downarrow y} = \mathbf{y}} p_x(\mathbf{x}) \,. \tag{2}$$

Note that if $y = \emptyset$, the marginal $p_x^{\downarrow y}$ is a (nonnegative real) number.

Partial ordering is given by weak Pareto dominance. Given two potentials p_x and q_x over Ω_x, we define $p_x \geq q_x$ if $p_x(\mathbf{x}) \geq q_x(\mathbf{x})$ for all $\mathbf{x} \in \Omega_x$. Note that if p_x and q_x have equal sum (i.e., $\sum_{\mathbf{x} \in \Omega_x} p_x(\mathbf{x}) = \sum_{\mathbf{x} \in \Omega_x} q_x(\mathbf{x})$) then $p_x \not\geq q_x$ and $q_x \not\geq p_x$ (unless $p_x = q_x$). This is the case, for example, of potentials representing (conditional) probability mass functions. Therefore, the identity $\mathcal{P}_x = \max(\mathcal{P}_x)$ holds for any set $\mathcal{P}_x$ of (conditional) probability mass functions. Let $\mathcal{P}$ denote the set of all probability potentials.

Proposition 4. *The system* $(\mathcal{P}, \mathcal{U}, \times, \downarrow, \leq)$ *is an ordered valuation algebra.*

Given a real number $\alpha > 1$, we define an equivalence relation $\equiv_\alpha$ over potentials such that any two potentials p_x and q_x are α-equivalent (i.e., $p_x \equiv_\alpha q_x$) if for all $\mathbf{x} \in \Omega_x$ either $p_x(\mathbf{x}) = q_x(\mathbf{x}) = 0$ or $p_x(\mathbf{x})$ and $q_x(\mathbf{x})$ are both positive and $\lfloor \log_\alpha p_x(\mathbf{x}) \rfloor = \lfloor \log_\alpha q_x(\mathbf{x}) \rfloor$.

2.3 Pairs of Potentials

The algorithms we develop in Sections 4 and 5 rely on a more abstract structure over pairs of potentials. Let $\phi_x = (p_x^\ell, p_x^r)$ denote a pair of probability potentials over x. The potentials p_x^ℓ and p_x^r are referred to as the left and right potentials of ϕ_x, respectively. For any two pairs of potentials ϕ_x and ψ_x, we define $\phi_x = (p_x^\ell, p_x^r) \geq (q_x^\ell, q_x^r) = \psi_x$ if $p_x^\ell \leq q_x^\ell$ and $p_x^r \geq q_x^r$. The partial order defined in this way reflects the nature of computations with credal networks. We seek for a solution that partly dominates (according to right potentials) all other potentials and partly is dominated by them (according to left potentials). It is in part this dichotomy in the objective that makes posterior inferences in credal networks much harder than their Bayesian counterpart.

If $\phi_x = (p_x^\ell, p_x^r)$ and $\phi_y = (p_y^\ell, p_y^r)$ are two pairs of potentials, we define their combination as the pair of left and right combinations of potentials, that is, $\phi_x \times \phi_y \triangleq (p_x^\ell \times p_y^\ell, p_x^r \times p_y^r)$. Similarly, the marginalization of a pair $\phi_x = (p_x^\ell, p_x^r)$ is performed on both potentials, $\phi_x^{\downarrow y} \triangleq ((p_x^\ell)^{\downarrow y}, (p_x^r)^{\downarrow y})$. Let Φ be the set of all pairs of potentials.

Proposition 5. *The system* $(\Phi, \mathcal{U}, \times, \downarrow, \leq)$ *is an ordered valuation algebra.*

Let 2^Φ and $\max(2^\Phi)$ denote, respectively, the set of all sets of pairs of potentials and the set of all sets of maximal pairs of potentials. It follows from Propositions 1 and 2 that the systems $(2^\Phi, \mathcal{U}, \otimes, \Downarrow)$ and $(\max(2^\Phi), \mathcal{U}, \oplus, \downdownarrows)$ are valuation algebras. Moreover, $\max$ is a homomorphism from 2^Φ to $\max(2^\Phi)$. Thus, given a collection of finite sets of pairs $\Psi_{x_1}, \ldots, \Psi_{x_n}$, we can obtain the set $\max(\Psi_y) \triangleq \max((\bigotimes \Psi_{x_i})^{\Downarrow y})$ of maximal marginal valuations potentially more efficiently by performing computations in the algebra of sets of maximal pairs, that is, by computing $\max((\bigoplus_i \max(\Psi_{x_i}))^{\downdownarrows y})$. Bentley et al. [5] showed that sets with n uniformly distributed pairs of potentials over a domain Ω_y have, on average, $O((\log n)^{2|\Omega_y|-1})$ maximal elements. Unfortunately, the uniformity assumption does not hold in the computations we perform, and we expect the average number of maximal elements to be higher than this. To our knowledge, it remains to be obtained any bounds or expectations on the size of maximal sets obtained from propagated valuations such as those generated by variable elimination. Note that, as with sets of probability potentials, if Ψ contains only valuations whose left or right potentials specify a probability mass function, then $\Psi = \max(\Psi)$.

We can have an upper bound on the cardinality of sets by relaxing the partial order to allow approximate Pareto dominance. Given a real number $\alpha > 1$, we define a relation $\leq_\alpha$ such that $\phi \leq_\alpha \psi$ denotes that by mistakenly assuming $\phi \leq \psi$ we introduce an error no greater than α in each coordinate. More formally, we define $\phi \leq_\alpha \psi$ if $(\alpha^{-1}, \alpha) \times \psi \geq \phi$. Note that $\leq_\alpha$ is neither transitive nor antisymmetric, and that we may have $\phi \leq_\alpha \psi$ for $\phi \not\leq \psi$.

The α-equivalence relation over potentials can easily be extended to pairs. Two pairs (p_x^ℓ, p_x^r) and (p_y^ℓ, p_y^r) are α-equivalent if $p_x^\ell \equiv_\alpha p_y^\ell$ and $p_y^r \equiv_\alpha p_y^r$. It is not difficult to see that $\phi \equiv_\alpha \psi$ implies both $\phi \leq_\alpha \psi$ and $\psi \leq_\alpha \phi$.

A $\leq_\alpha$-covering for a set of pairs of potentials Ψ_x provides an approximated version of Ψ_x, one in which for each $\phi_x \in \Psi_x$ we are guaranteed to have a pair ψ_x in the covering such that the left and right potentials of ψ_x and ϕ_x differ in each coordinate by a factor no greater than α. We can easily obtain a $\leq_\alpha$-covering of Ψ_x of bounded cardinality from its quotient set Ψ_x/α, that is, by discarding one of any two α-equivalent pairs in Ψ_x. The approximation algorithm we develop in Section 5 strongly relies on the following results.

Lemma 6. *If $k_1, \ldots, k_m$ are positive integers and $\Psi_{x_1}, \Psi'_{x_1}, \ldots, \Psi_{x_m}, \Psi'_{x_m}$ are set valuations such that for $i = 1, \ldots, m$ Ψ'_{x_i} is a $\leq_{\alpha^{k_i}}$-covering for Ψ_i, then $\Psi'_{x_1} \otimes \cdots \otimes \Psi'_{x_m}$ is a $\leq_\beta$-covering for $\Psi_{x_1} \otimes \cdots \otimes \Psi_{x_m}$, where $\beta = \alpha^{\sum_{i=1}^m k_i}$.*

Proof. We work by induction on $j = 1, \ldots, m$. For $j = 1$, it follows directly that Ψ'_1 is a $\leq_{\alpha^{k_1}}$-covering for Ψ_1. Assume the result holds for $1 \leq j < m - 1$, and consider any pair $\phi = \phi' \times \phi''$ in $\Psi_{x_1} \otimes \cdots \otimes \Psi_{x_{j+1}}$, where $\phi' \in \Psi_{x_1} \otimes \cdots \otimes \Psi_{x_j}$ and $\phi'' \in \Psi_{x_{j+1}}$. There is $\psi = \psi' \times \psi''$ in $\Psi'_{x_1} \otimes \cdots \otimes \Psi'_{x_{j+1}}$, where $\psi' \in \Psi'_{x_1} \otimes \cdots \otimes \Psi'_{x_j}$ and $\psi'' \in \Psi_{x_{j+1}}$, such that $(\alpha^{-\sum_{i=1}^j k_i}, \alpha^{\sum_{i=1}^j k_i}) \times \psi' \geq \phi'$ (by assumption) and $(\alpha^{-k_{j+1}}, \alpha^{k_{j+1}}) \times \psi'' \geq \phi''$. It follows from (A4) that $(\alpha^{-\sum_{i=1}^{j+1} k_i}, \alpha^{\sum_{i=1}^{j+1} k_i}) \times \psi \geq \phi$.□

Let $\Psi_{x_1}, \ldots, \Psi_{x_m}$ denote sets of pairs of potentials which take values on the interval $[0, 1]$, and let b be the number of bits required to encode these sets.

Proposition 7. *The number of elements in $(\Psi_{x_1} \otimes \cdots \otimes \Psi_{x_m})^{\Downarrow y}/\alpha$ is $O((bm\alpha/(\alpha - 1))^{2|\Omega_y|})$.*

The latter result is in fact an adaptation of Papadimitriou and Yannakakis' result on the boundedness of ϵ-approximate Pareto curves in multi-objective optimization problems [1, Theorem 1].

3 Credal Networks

In this section we review the basic concepts and computational challenges of extensively specified credal networks. Let $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ be a DAG, and X a node in $\mathcal{U}$. We write $\mathrm{pa}(X) \triangleq \{Y \in \mathcal{U} : (Y, X) \in \mathcal{E}\}$ to denote the parents of X, $\mathrm{ch}(X) \triangleq \{Y \in \mathcal{U} : (X, Y) \in \mathcal{E}\}$ to denote the children of X in $\mathcal{U}$, and $\mathrm{fa}(X) \triangleq \{X\} \cup \mathrm{pa}(X)$ to denote the family of X. We call Y a descendant of X if there is a directed path from X to Y in $\mathcal{G}$.

An *extensive credal set* K_x is a set of probability potentials p_x over domain Ω_x. Given an extensive credal set K_x, we write $\mathrm{H}(K_x)$ to denote its convex hull (i.e., the set obtained by all convex combinations of elements in K_x), and $\mathrm{ext}[\mathrm{H}(K_x)]$ to denote its extreme points (i.e., the elements of $\mathrm{H}(K_x)$ that cannot be written as a convex combination of other elements). The convex hull of a set and the set of its extreme points are themselves extensive credal sets.

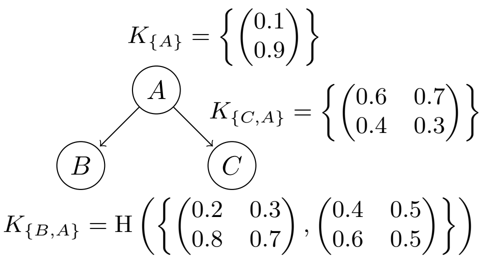
$K_{\{A\}} = \left\{ \begin{pmatrix} 0.1 \\ 0.9 \end{pmatrix} \right\}$
A
$K_{\{C,A\}} = \left\{ \begin{pmatrix} 0.6 & 0.7 \\ 0.4 & 0.3 \end{pmatrix} \right\}$
B
C
$K_{\{B,A\}} = \mathrm{H}\left(\left\{ \begin{pmatrix} 0.2 & 0.3 \\ 0.8 & 0.7 \end{pmatrix}, \begin{pmatrix} 0.4 & 0.5 \\ 0.6 & 0.5 \end{pmatrix} \right\} \right)$

Figure 1: Example of extensively specified credal network.

An *extensively specified credal network* is a pair $(\mathcal{G}, \mathbb{K})$, where $\mathbb{K}$ is a collection of finitely-generated closed convex extensive credal sets $K_{\mathrm{fa}(X)}$, one for each $X \in \mathcal{U}$, such that each potential $p_{\mathrm{fa}(X)} \in K_{\mathrm{fa}(X)}$ satisfies $\sum_{\mathbf{x}^{\downarrow \mathrm{pa}(X)} = \pi} p_{\mathrm{fa}(X)}(\mathbf{x}) = 1$ for all $\pi \in \Omega_{\mathrm{pa}(X)}$ (i.e., they represent conditional probability mass functions $p(X|\,\mathrm{pa}(X))$). Figure 1 depicts a simple extensively specified credal network over 3 binary-valued variables.

The *strong extension* of a credal network is given by the credal set generated by the convex closure of the product of all extensive credal sets in $\mathbb{K}$,

$$K_{\mathcal{U}}^{\text{strong}} \triangleq \mathrm{H}\left(\bigotimes_{X \in \mathcal{U}} K_{\mathrm{fa}(X)} \right) . \qquad (3)$$

Since the product of local extremes $K_{\mathcal{U}}^{\text{ext}} \triangleq \bigotimes_{X \in \mathcal{U}} \mathrm{ext}[K_{\mathrm{fa}(X)}]$ is a subset of the strong extension (by definition), we have that $\mathrm{ext}[K_{\mathcal{U}}^{\text{strong}}] = \mathrm{ext}[\mathrm{H}(K_{\mathcal{U}}^{\text{ext}})] \subseteq K_{\mathcal{U}}^{\text{ext}}$. Notice that $K_{\mathcal{U}}^{\text{ext}}$ contains a finite number of elements.

Let $q, e \subset \mathcal{U}$ denote disjoint sets of query and evidence variables, respectively, and $(\mathbf{q}, \mathbf{e})$ an element of $\Omega_{q \cup e}$. Inference with credal networks consists in computing lower and upper posterior probabilities (we assume $p^{\downarrow e}(\mathbf{e}) > 0$ for all $p \in K_{\mathcal{U}}^{\text{strong}}$):

$$\underline{p}(\mathbf{q}|\mathbf{e}) \triangleq \min_{p \in K_{\mathcal{U}}^{\text{strong}}} \frac{p^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e})}{p^{\downarrow e}(\mathbf{e})} \, , \qquad (4)$$

$$\overline{p}(\mathbf{q}|\mathbf{e}) \triangleq \max_{p \in K_{\mathcal{U}}^{\text{strong}}} \frac{p^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e})}{p^{\downarrow e}(\mathbf{e})} \, . \qquad (5)$$

Our goal in the rest of this section is to show that the continuous optimizations of Equations (4) and (5) can be mapped into problems of computing maximal sets of marginals of the combinations of finite sets of pairs of potentials. We begin with a well-known result that the solutions to the convex optimizations in Equation (5) are attained at extreme points of the strong extension [18]. Since any non-extreme point of $K_{\mathcal{U}}^{\text{ext}}$ is also a non-extreme point of the strong extension, we have that

$$\overline{p}(\mathbf{q}|\mathbf{e}) = \max_{p \in K_{\mathcal{U}}^{\text{ext}}} \frac{p^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e})}{p^{\downarrow e}(\mathbf{e})} \tag{6}$$

$$= \max_{p_{\mathcal{U}} \in K_{\mathcal{U}}^{\text{ext}}} \frac{p^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e})}{p^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e}) + p^{\downarrow q \cup e}(\neg \mathbf{q}, \mathbf{e})}\,, \tag{7}$$

where $p^{\downarrow q \cup e}(\neg \mathbf{q}, \mathbf{e}) \triangleq \sum_{\mathbf{q}' \in \Omega_q : \mathbf{q}' \neq \mathbf{q}} p^{\downarrow q \cup e}(\mathbf{q}', \mathbf{e})$. We can derive analogous equations for the lower bound. The passage from Equation (6) to (7) follows from the definition of marginalization. Notice that Equation (7) states a combinatorial problem over products of local extreme points. If $\underline{p}(\mathbf{q}|\mathbf{e}) > 0$, we can divide the numerator and the denominator of Equation (7) by $p^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e}) > 0$ and obtain

$$\overline{p}(\mathbf{q}|\mathbf{e}) = \max_{p \in K_{\mathcal{U}}^{\text{ext}}} \left(1 + \frac{p^{\downarrow q \cup e}(\neg \mathbf{q}, \mathbf{e})}{p^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e})} \right)^{-1}. \tag{8}$$

For any potential $p \in K_{\mathcal{U}}^{\text{ext}}$, let $p_{\mathbf{q}|\mathbf{e}}$ denote the posterior probability obtained by p, that is, $p_{\mathbf{q}|\mathbf{e}} \triangleq [1 + p^{\downarrow q \cup e}(\neg \mathbf{q}, \mathbf{e}) / p^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e})]^{-1}$. Now consider two potentials p and r such that $p^{\downarrow q \cup e}(\neg \mathbf{q}, \mathbf{e}) \leq r^{\downarrow q \cup e}(\neg \mathbf{q}, \mathbf{e})$ and $p^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e}) \geq r^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e})$. Clearly, $p_{\mathbf{q}|\mathbf{e}} \geq r_{\mathbf{q}|\mathbf{e}}$, and r is not a solution of the maximization problem (conversely, p is not a solution of the minimization problem). This allows us to define a partial ordering among solutions $p \in K_{\mathcal{U}}^{\text{ext}}$.

Let $\Phi_{\mathbf{q}|\mathbf{e}}$ denote the set of pairs of potentials $(p^{\downarrow q \cup e}(\neg \mathbf{q}, \mathbf{e}), p^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e}))$, where $p \in K_{\mathcal{U}}^{\text{ext}}$. Then Equation (8) can be rewritten as

$$\overline{p}(\mathbf{q}|\mathbf{e}) = \max_{(p^\ell, p^r) \in \max(\Phi_{\mathbf{q}|\mathbf{e}})} \left(1 + p^\ell / p^r \right)^{-1}. \tag{9}$$

Basically, what Equation (9) states is that we can narrow down the optimization space to the set of potentials whose corresponding pairs in $\Phi_{\mathbf{q}|\mathbf{e}}$ are not smaller than any other pair in the set (conversely, we take the set of minimal elements in the minimization case). Although this set could be as large as $K_{\mathcal{U}}^{\text{ext}}$, our experiments show that most often it is significantly smaller. Thus, if $\max(\Phi_{\mathbf{q}|\mathbf{e}})$ is sufficiently small, we can find the solution by a simple enumerative scheme, and the optimization problem is then converted into the problem of computing the maximal elements of $\Phi_{\mathbf{q}|\mathbf{e}}$, which can be done by the variable elimination procedure in Algorithm 1, as the following section shows.

4 Exact Inference

In this section we describe an algorithm for exact computation of upper posterior probabilities in credal networks. An algorithm for obtaining lower probabilities can be obtained in a very similar way.

For any variable X and a subset $\mathcal{X} \subset \Omega_X$, we define the identity potential $I_{\mathcal{X}}$ as a potential over X that returns 1 for $\mathbf{x} \in \mathcal{X}$ and 0 otherwise. If $\mathcal{X} = \{\mathbf{x}\}$ is a singleton, we write $I_{\mathbf{x}}$. For any $\mathbf{x} \in \Omega_X$, we define the set $\neg \mathbf{x} \triangleq \Omega_X \setminus \{\mathbf{x}\}$.

Consider a credal network $(\mathcal{G}, \mathbb{K})$, an elimination ordering $o = (X_1, \ldots, X_n)$ of the variables in $\mathcal{U}$, sets of query and evidence variables q and e, and a query-evidence pair $(\mathbf{q}, \mathbf{e}) \in \Omega_{q \cup e}$. The variable elimination algorithm (Algorithm 1) can be used to compute exact upper posterior probabilities using the valuation algebra of sets of maximal pairs of potentials in the following way. Let Ψ be the set that contains (i) for each $X \in \mathcal{U}$ a set-valuation $\Psi_X \triangleq \{(p_{\text{fa}(X)}, p_{\text{fa}(X)}) : p_{\text{fa}(X)} \in \text{ext}[K_{\text{fa}(X)}]\}$ in Φ; (ii) a set-valuation $\Psi_q \triangleq \{(I_{\neg \mathbf{q}}, I_{\mathbf{q}})\}$ in Ψ; and (iii) for each $E \in e$ a set-valuation $\Psi_E \triangleq \{(I_{\mathbf{e}^{\downarrow E}}, I_{\mathbf{e}^{\downarrow E}})\}$ in Ψ. Let Γ be the output of the variable elimination algorithm with max-combination, and max-marginalization and inputs Ψ, $y = \emptyset$ and ordering o, and let $p_{\mathbf{q}|\mathbf{e}} \triangleq \max_{(p^\ell, p^r) \in \Gamma} \left(1 + p^\ell / p^r\right)^{-1}$. Finally, let $\overline{p}(\mathbf{q}|\mathbf{e})$ be the solution of the maximization problem in Equation (5). The following result states the correctness of the upper posterior probability obtained the procedure.

Theorem 8. $p_{\mathbf{q}|\mathbf{e}} = \overline{p}(\mathbf{q}|\mathbf{e})$.

Proof. The sets $\Psi_X, \Psi_q, \Psi_E \in \Psi$ as well as the sets Ψ^i generated by the variable elimination algorithm are valuations in the valuation algebra of sets of maximal pairs of potentials. It follows from (A1)–(A3) that

$$\Gamma = \left(\Psi_q \bigoplus_{E \in e} \Psi_E \bigoplus_{X \in \mathcal{G}} \Psi_X \right)^{\Downarrow \emptyset} \tag{10}$$

$$= \max\left(\left[\Psi_q \bigotimes_{E \in e} \Psi_E \bigotimes_{X \in \mathcal{G}} \Psi_X \right]^{\downarrow \emptyset} \right), \tag{11}$$

where the last equivalence is obtained by repeatedly applying Proposition 3. Recall that combination of pairs is defined as the pair formed by the combination of left potentials and the combination of right potentials. Therefore, Γ is a set of maximal pairs of potentials (p^ℓ, p^r), where by definition of Ψ_q, Ψ_E, and Ψ_X,

$$p^\ell = \left(I_{\neg \mathbf{q}} \bigotimes_{E \in e} I_{\mathbf{e}^{\downarrow E}} \bigotimes_{X \in \mathcal{G}} p_{\text{fa}(X)} \right)^{\downarrow \emptyset} \tag{12}$$

$$= p_{\mathcal{U}}^{\downarrow q \cup e}(\neg \mathbf{q}, \mathbf{e})\,, \tag{13}$$

$$p^r = \left(I_{\mathbf{q}} \bigotimes_{E \in e} I_{\mathbf{e}^{\downarrow E}} \bigotimes_{X \in \mathcal{G}} p_{\text{fa}(X)} \right)^{\downarrow \emptyset} \tag{14}$$

$$= p_{\mathcal{U}}^{\downarrow q \cup e}(\mathbf{q}, \mathbf{e})\,. \tag{15}$$

Moreover, p^ℓ and p^r are compatible, that is, for any potential $p_{\text{fa}(X)}$ in p^ℓ taken from a local extensive credal set $K_{\text{fa}(X)}$, the same potential appears in p^r and no other potential from $K_{\text{fa}(X)}$. Hence, $\Gamma = \max(\Phi_{\mathbf{q}|\mathbf{e}})$. The result is obtained by comparing the definition of $p_{\mathbf{q}|\mathbf{e}}$ and Equation (9). □

The complexity of the algorithm is upper bounded by the cost of the combination of sets of pairs in computing Ψ^i during the variable elimination part. Each of these computations takes time polynomial in the size of the largest set, which might be exponential in the size of the input sets. For instance, the size of the largest potential is a function of the topology of $\mathcal{G}$ and the given elimination ordering o. The number of elements of a set, on the other hand, depends on the number of non-maximal elements that are discarded at each combination or marginalization operation. In the worst-case scenario where no element is ever discarded, the algorithm runs in exponential time even if the network treewidth and the cardinality of the frames of the input sets are bounded (which is not surprising given that the problem is NP-hard under such assumptions).

An algorithm for lower posterior probabilities can be obtained by substituting sets of maximal valuations and maximizations by sets of minimal valuations and minimizations, respectively. The correctness and complexity analyses are analogous to the maximization case.

5 FPTAS

The computational bottleneck of the variable elimination procedure presented in Section 4 is the existence of large sets at some point in the propagation step (apart from the inherent difficulty of manipulating potentials over large domains). We can remedy the large set problem by trading off accuracy and running time. In this section, we devise a multiplicative approximation scheme that runs in time polynomial in the number of potentials of the input extensive credal sets, but it is still exponential in the size of the largest pair ψ^{X_i} generated during the propagation step, which depends only on the sizes of the frames of the variables and the network treewidth. For the rest of this section, we assume the size of variable frames and the network treewidth to be bounded by a constant. Additionally, we require the input potentials to be represented by rational numbers, so that the length of the input is well-defined. The approximation scheme we obtain is an FPTAS, that is, a family of algorithms parameterized by $\epsilon > 0$ that returns in time polynomial to $1/\epsilon$ and to the input size a feasible solution that is no worse than the optimal solution by a factor of ϵ. If x^* is the optimal solution (of a maximization problem), the approximation algorithm returns a solution x such that $x^*/(1+\epsilon) \leq x \leq x^*$.

Given a real number α greater than one, we define the α-combination of two set-valuations Ψ_x and Ψ_y as the quotient set of the their set-combination, that is, $\Psi_x \boxtimes_\alpha \Psi_y \triangleq (\Psi_x \otimes \Psi_y)/\alpha$. The operation $\boxtimes_\alpha$ is not associative, that is, there are set-valuations Ψ_x, Ψ_y and Ψ_z such that $(\Psi_x \boxtimes_\alpha \Psi_y) \boxtimes_\alpha \Psi_z$ differs from $\Psi_x \boxtimes_\alpha (\Psi_y \boxtimes_\alpha \Psi_z)$. Nevertheless, the order in which sets are α-combined does not alter the combined approximation factor, as the following result states.

Lemma 9. *If $\Psi_1, \ldots, \Psi_m$ are set-valuations, then $\Psi_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_m$ (where the operations are applied in any order) is a $\leq_\beta$-covering for $\Psi_1 \otimes \cdots \otimes \Psi_m$, where $\beta = \alpha^{m-1}$.*

Proof. We work by induction on $k = 2, \ldots, m$. For $k = 2$, it follows directly from the definition of α-combination that $\Psi_1 \boxtimes_\alpha \Psi_2$ is an $\leq_\alpha$-combination for $\Psi_1 \otimes \Psi_2$. Assume for $k \in \{2, \ldots, m-1\}$ that $\Psi_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_{k-1}$ is a $\leq_\beta$-covering for $\Psi_1 \otimes \cdots \otimes \Psi_{k-1}$, where $\beta = \alpha^{k-2}$. Consider any pair $\phi = \phi' \times \phi''$ in $\Psi_1 \otimes \cdots \otimes \Psi_k$, where $\phi' \in \Psi_1 \otimes \cdots \otimes \Psi_{k-1}$ and $\phi'' \in \Psi_k$. There is $\psi = \psi' \times \psi''$ in $\Psi_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_{k-1} \otimes \Psi_k$, where $\psi' \in \Psi_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_{k-1}$ and $\psi'' \in \Psi_k$, such that $\psi' \geq_\beta \phi'$ (by assumption) and $\psi'' = \phi''$. Then it follows from (A4) that $\psi \geq_\beta \phi$, or equivalently that $(\beta^{-1}, \beta) \times \psi \geq \phi$. But since $\Psi_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_k$ is a $\leq_\alpha$-covering for $\Psi_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_{k-1} \otimes \Psi_k$, there is $\psi''' \in \Psi_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_k$ such that $\psi''' \geq_\alpha \psi$, or equivalently that $(\alpha^{-1}, \alpha) \times \psi''' \geq \psi$. By combining both sides with (β^{-1}, β) and applying (A4) we get to

$$(\beta^{-1}, \beta) \times (\alpha^{-1}, \alpha) \times \psi''' \geq (\beta^{-1}, \beta) \times \psi \geq \phi \,,$$

and hence $(\alpha^{-(k-1)}, \alpha^{k-1}) \times \psi''' \geq \phi$, and $\psi''' \geq_{\alpha^{k-1}} \phi$. The lemma follows from the induction. □

Thus, by properly choosing the value of α we can obtain a covering that approximates a combination of set-valuations with errors as small as we want. In addition, Proposition 7 guarantees that the sets obtained after each α-combination have cardinality polynomial in the input length and in the maximum error, and so the covering.

We can then modify the exact variable elimination algorithm devised in Section 4 to provide an FPTAS by substituting max-combination and max-marginalization by α-combination with $\alpha = 1 + \epsilon/4n$ and set-marginalization. Let Ψ^i and Ψ^i_α denote, respectively, the sets obtained in the ith iteration of the loop step of variable elimination using set-combination and α-combinations (and both with set-marginalization). In other words, Ψ^i is the set obtained by a brute-force elimination algorithm, whereas Ψ^i_α denote the sets obtained by the approximation algorithm. Similarly, we let Γ and Γ_α denote the outputs of variable elimination with set-combination and α-combination, respectively.

Let s_1 denote the number of set-valuations that are combined to compute Ψ^1_α (and also Ψ^1) minus one, that is, $s_1 \triangleq |\mathcal{B}_1| - 1$. Then, for $i = 2, \ldots, n$, we define s_i recursively as $s_i \triangleq |\mathcal{B}_i| - 1 + \sum_{j:\Psi^j_\alpha \in \mathcal{B}_i} s_j$. Intuitively, s_i denote the number of valuations from the input that are required either directly or indirectly to compute Ψ^i_α (and also Ψ^i) minus one. Hence, if Ψ is the set obtained after the loop step, we have that $|\Gamma_\alpha| + \sum_{i:\Psi^i_\alpha \in \Psi} s_i = n$, since there are n set-valuations given as input and each is used exactly once in the computation of some Ψ^i_α (or Ψ^i).

The following lemma relates the set-valuations propagated by variable elimination with α-combination to the corresponding sets obtained by set-combination.

Lemma 10. *For $i = 1, \dots, n$, the set-valuation Ψ^i_α is a $\leq_{\alpha^{s_i}}$-covering for Ψ^i.*

Proof. For $i = 1$ the result follows directly from Lemma 9. Without loss of generality, let $\Psi^i = [\Psi_1 \otimes \cdots \otimes \Psi_k \otimes \cdots \otimes \Psi_{|\mathcal{B}_i|}]^{-X_i}$, where $\Psi_1, \dots, \Psi_k$ denote set-valuations given as input and $\Psi_{k+1}, \dots, \Psi_{|\mathcal{B}_i|}$ denote sets Ψ^j $(j < i)$ generated in the propagation step. Similarly, let $\Psi^i_\alpha = [\Psi_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_k \boxtimes_\alpha \Psi'_{k+1} \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi'_{|\mathcal{B}_i|}]^{-X_i}$, where, for $k + 1 < \ell < |\mathcal{B}_i|$, $\Psi_\ell = \Psi^j$ implies $\Psi'_\ell = \Psi^j_\alpha$. Assume by induction that the result holds for $1, \dots, i - 1$. Hence, if $\Psi'_\ell = \Psi^j_\alpha$ then Ψ'_ℓ is a $\leq_{\alpha^{s_j}}$-covering for Ψ_ℓ. Now, consider any pair $\phi = [\phi' \times \phi'']^{-X_i} \in \Psi^i$, where $\phi' \in \Psi_1 \otimes \cdots \otimes \Psi_k$ and $\phi'' \in \Psi_{k+1} \otimes \cdots \otimes \Psi_{|\mathcal{B}_i|}$. From Lemma 9, we have that there is $\psi' \in \Psi_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_k$ such that $(\alpha^{-k+1}, \alpha^{k-1}) \times \psi' \geq \phi'$. Likewise, since $\Psi'_{k+1} \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi'_{|\mathcal{B}_i|}$ is a $\leq_{\alpha^{|\mathcal{B}_i|-(k+1)}}$-covering for $\Psi'_{k+1} \otimes \cdots \otimes \Psi'_{|\mathcal{B}_i|}$ (by Lemma 9) and $\Psi'_{k+1} \otimes \cdots \otimes \Psi'_{|\mathcal{B}_i|}$ is a $\leq_{\alpha^{\sum_\ell s_\ell}}$-covering for $\Psi_{k+1} \otimes \cdots \otimes \Psi_{|\mathcal{B}_i|}$ (by Lemma 6 and the induction hypothesis), there is $\psi'' \in \Psi'_{k+1} \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi'_{|\mathcal{B}_i|}$ such that $(\alpha^{-s_i+k}, \alpha^{s_i-k}) \times \psi'' \geq \phi''$. Since $\equiv_\alpha$ implies $\leq_\alpha$, there is $\psi \in (\Psi_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_k) \boxtimes_\alpha (\Psi_{k+1} \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi_{|\mathcal{B}_i|})$ such that $(\alpha^{-1}, \alpha) \times \psi \geq \psi' \times \psi''$. Thus, it follows from (A4) that $[(\alpha^{-s_i}, \alpha^{s_i}) \times \psi]^{-X_i} \geq \phi$. But from (A3) we have that $[(\alpha^{-s_i}, \alpha^{s_i}) \times \psi]^{-X_i} = (\alpha^{-s_i}, \alpha^{s_i}) \times \psi^{-X_i}$, where $\psi^{-X_i} \in \Psi^i_\alpha$. Since this is true for any $\phi \in \Psi^i$, the result holds for i. The lemma follows from the induction. □

Consider a credal network $(\mathcal{G}, \mathbb{K})$, an elimination ordering $o = (X_1, \dots, X_n)$ of the variables in $\mathcal{U}$, sets of query and evidence variables q and e, and a query-evidence pair $(\mathbf{q}, \mathbf{e}) \in \Omega_{q \cup e}$. Let Ψ be a collection of sets of pairs as defined in Section 4, and consider the variable elimination algorithm with inputs Ψ, $y = \emptyset$ and o, and α-combination and set-marginalization. Finally, return $p_{\mathbf{q}|\mathbf{e}} \triangleq \max_{(p^\ell, p^r) \in \Gamma_\alpha} (1 + p^\ell / p^r)^{-1}$ as the approximate solution output.

Theorem 11. *The procedure described is an FPTAS for computing upper posterior probabilities for networks of bounded treewidth and number of states per variable.*

Proof. First, we analyze the time complexity of the algorithm. We are thus interested in the maximum cardinality of a set Ψ^i_α, and in the cardinality of the domain of a valuation generated in the loop step. The boundedness assumptions imply that the cardinality of the domain of any propagated valuation is smaller than a constant. Hence, the polynomial time complexity depends on $|\Psi^i_\alpha|$ being bounded. For $i = 1, \dots, n$, any valuation $\phi^i \in \Psi^i_\alpha$ is produced by first combining valuations that are either in some previously generated set Ψ^j_α $(j < i)$ or in a set given as input, and then eliminating X_i from it. Thus, by recursively applying (A1)–(A3) to factorize each valuation from a Ψ^j_α into a combination of valuations and moving the eliminations out, we have that $\phi^i = [\phi_1 \times \cdots \times \phi_{s_i+1}]^{-\{X_1, \dots, X_i\}}$, where each ϕ_j is in a set-valuation given as input. Hence, each Ψ^i_α can be factorized as $[\Psi_1 \otimes \cdots \otimes \Psi_{s_i}]^{-\{X_1, \dots, X_i\}}$, where each Ψ_i is a subset of a set-valuation given as input. It follows then from Proposition 7 that Ψ^i_α has $O([bs_i \alpha / (\alpha - 1)]^{2\omega})$, where ω is a constant greater than the cardinality of the domain of any ϕ^i. Since $\alpha = 1 + \epsilon / 4n$, $O([bs_i \alpha / (\alpha - 1)]^{2\omega}) \leq O((4n^2 b / \epsilon)^{2\omega})$, where b is the length of the input in bits. Therefore the algorithm runs in time polynomial in the input, in the given approximation factor ϵ, and in the number of variables n.

Let $\overline{p}(\mathbf{q}|\mathbf{e}) \triangleq \max_{(p^\ell_*, p^r_*) \in \Gamma} (1 + p^\ell_* / p^r_*)^{-1}$ denote the optimum value. We now show that the approximation algorithm yields a solution such that $p_{\mathbf{q}|\mathbf{e}} \geq \overline{p}(\mathbf{q}|\mathbf{e}) / (1 + \epsilon)$ for any given positive ϵ. Let $\Psi'_1, \dots, \Psi'_m$ denote the sets Ψ^i_α in Ψ after the loop step of the approximation algorithm, where $m = |\Gamma_\alpha|$, and let $\Psi_1, \dots, \Psi_m$ be the sets Ψ^i in Ψ after the loop step of the brute-force version. Then, $\Gamma_\alpha = \Psi'_1 \boxtimes_\alpha \cdots \boxtimes_\alpha \Psi'_m$ and $\Gamma = \Psi_1 \otimes \cdots \otimes \Psi_m$. It follows from Lemma 9 that Γ_α is a $\leq_{\alpha^{m-1}}$-covering for $\Psi'_1 \otimes \cdots \otimes \Psi'_m$, which in turn is a $\leq_{\alpha^{n-m}}$-covering for Γ, by Lemma 10. Hence, for any $\phi \in \Gamma$ there is $\psi \in \Gamma_\alpha$ such that $(\alpha^{-(n-1)}, \alpha^{n-1}) \times \psi \geq \phi$ an thus $(\alpha^{-n}, \alpha^n) \times \psi \geq \phi$. In particular, there is $\psi = (p^\ell, p^r) \in \Gamma_\alpha$ such that $\psi \geq_{\alpha^n} (p^\ell_*, p^r_*) = \phi^*$. Therefore, $p^\ell \leq \alpha^n p^\ell_*$, $\alpha^n p^r \geq p^r_*$, and

$$\begin{aligned}(1 + p^\ell / p^r)^{-1} &\geq (1 + \alpha^{2n} p^\ell_* / p^r_*)^{-1} \\ &\geq \alpha^{-2n} (1 + p^\ell_* / p^r_*)^{-1} \,.\end{aligned}$$

Since $\alpha = (1 + \epsilon / 4n)$, we have that

$$\begin{aligned}(1 + p^\ell / p^r)^{-1} &\geq (1 + \epsilon / 4n)^{-2n} (1 + p^\ell_* / p^r_*)^{-1} \\ &\geq (1 + \epsilon)^{-1} (1 + p^\ell_* / p^r_*)^{-1} \\ &= (1 + \epsilon)^{-1} \overline{p}(\mathbf{q}|\mathbf{e}) \,,\end{aligned}$$

where the second passage is due to the inequality $(1 + x/z)^z \leq 1 + 2x$, valid for any $x \in [0, 1]$ and any positive integer z. Hence, $p_{\mathbf{q}|\mathbf{e}} \geq \overline{p}(\mathbf{q}|\mathbf{e}) / (1 + \epsilon)$. □

Finally, we note that the approximation algorithm can be made more efficient by discarding non-maximal pairs from sets Ψ^i_α like in the exact algorithm in Section 4. This is done in our implementation of the algorithm whose performance we evaluate in the next section.

6 Experiments

We evaluate the performance of the exact and the approximation algorithms on a collection of extensively specified credal networks randomly generated using the BNGen package [14]. The graph topology of these networks is divided in three types, namely (from the simplest to the

Type	Exact Method				Approx. ($\epsilon = 0.1$)				Integer Programming			
	%	Time(sec)			%	Time(sec)			%	Time(sec)		
	solved	Median	Avg.	SD	solved	Median	Avg.	SD	solved	Median	Avg.	SD
M10-2-16	20	824	5617	9923	21	955	6978	11157	6	40464	35079	10451
M10-2-2	100	0.04	0.04	0.03	100	0.04	0.04	0.03	100	2	6	8
M10-2-4	100	4	1096	3906	100	3	276	1025	73	11445	13487	9206
M10-4-2	100	0.19	0.38	0.46	100	0.2	0.41	0.49	75	1320	5699	8922
M10-4-4	100	248	2030	4407	100	238	1992	4335	3	8459	8459	1108
M20-2-2	95	113	1835	4304	96	95	1592	3747	46	8039	12601	10654
M20-4-2	81	1154	5864	9584	81	1266	6009	9594	0	–	–	–
M30-2-2	26	8560	12170	11710	30	4032	13775	13734	3	9484	9484	0
P10-4-16	10	15428	16877	14159	10	16719	16470	13080	0	–	–	–
P10-4-2	100	0.04	0.04	0.03	100	0.04	0.05	0.03	96	248	2451	7752
P10-4-4	100	4	1977	5075	100	4	2095	5476	6	15101	15101	1564
P20-4-2	100	39	2055	5097	100	32	1691	4483	0	–	–	–
P20-4-4	6	20669	20669	20588	6	13484	13484	13400	36	5393	8931	6016
P30-4-2	6	8207	8207	1385	6	5171	5171	1306	0	–	–	–
T10-4-16	13	1559	1381	687	16	1855	9778	16704	0	–	–	–
T10-4-2	100	0.04	0.04	0.02	100	0.04	0.05	0.02	100	12	14	7
T10-4-4	100	6	784	3554	100	6	674	3129	0	–	–	–
T20-4-2	96	89	2415	6164	96	73	2597	7009	13	29022	29587	4839

Table 2: Performance of proposed methods and the integer programming idea. Columns show percentage of solved cases, median, mean and standard deviation (SD) for each group. Numbers greater than one are truncated.

most complicated): trees (graphs with maximum in-degree one), polytrees (graphs where the underlying undirected graph has no cycles), and multi-connected (DAGs without restrictions). All networks have treewidth no greater than four, 10 to 30 nodes, 2 to 4 states per variable, and 2 to 16 potentials in each local extensive credal set. In order to have statistically significant measures, we group networks of similar structure which we identify by the notation *Sn-k-c*, where S is one of T (for trees), P (for polytrees), or M (for multi-connected), n is the number of nodes in the graph, k is the number of states per node, and c is the cardinality of the credal sets. The number of networks in each group is either 30 or 60 (see second column of Table 3). For each network, we set some evidence to every leaf node and arbitrarily choose a node with no parents as query. This creates problems where a brute-force approach would have to execute c^n Bayesian network inferences. The elimination ordering is obtained by a greedy algorithm that attempts to minimize the size of propagated set-valuations. To make the removal of non-maximal valuations effective, we ensure the set-valuation Ψ_q is in $\mathcal{B}_1$, even if it is not required (i.e., if $X_1 \notin q$). Since the query has no parents, this can only increase the treewidth by one.

Table 2 reports the performance of the exact and the approximation algorithms along with the integer programming method of de Campos and Cozman [8]. The latter is a state-of-the-art solver for inference in credal networks that performs a symbolic inference in the credal network to obtain a set of linear constraints over continuous and binary optimization variables, which is then processed by a mixed integer programming solver. For each inference method and network group, Table 2 contains the percentage of cases that were correctly solved using at most 12 hours of CPU time and 2GB of RAM, and the median, average and standard deviation of the time spent. Regarding the mixed integer programming, we considered an instance solved only if the lower and upper bounds given returned by the solver matched. As the networks become more complicated, the percentage of solved cases reduces and the time to solve each case increases. The superiority against the integer programming is clear, though we suspect the integer programming might be suffering from numerical issues that are preventing it to achieve better results. Regarding the approximation, we see no significant reduction in time nor increase in the number of solved cases with respect to the exact method. Some facts contribute to that: (i) the limit of 12 hours of computation might be too short to get a consistent difference in the performance of the methods; (ii) the approximation has an additional computational cost in removing α-equivalent pairs, which is asymptotically irrelevant but significant otherwise; (iii) the number of discarded potentials in each step depends on the elimination order, the dimension of the potentials, and the randomness of input values. Table 3 shows average and standard deviation of the maximum number of elements in a set generated by the exact and approximation algorithms in the loop step. Recall that the complexity is related to the number of elements (as well as the cardinality) of the set-valuations generated. For instance, there would eventually be c^n propagated potentials if no $\leq$ relation (conversely, $\leq_\alpha$ relation) was observed.

Type	# of	Exact		Approximation	
	nets	Avg.	SD	Avg.	SD
M10-2-16	60	36046	28928	34579	28563
M10-2-2	60	154	141	130	109
M10-2-4	60	24642	64632	7254	9439
M10-4-2	60	225	128	224	127
M10-4-4	60	46147	65056	42664	55941
M20-2-2	60	37515	61606	28977	46774
M20-4-2	60	67573	73868	66185	73362
M30-2-2	30	93213	55519	81624	57996
P10-4-16	30	104468	75687	92784	64183
P10-4-2	30	115	100	114	100
P10-4-4	30	37155	78008	31361	64117
P20-4-2	30	24856	44469	20337	37219
P20-4-4	30	76083	68966	58358	51241
P30-4-2	30	92744	5476	65708	16654
T10-4-16	30	11840	9570	11834	9572
T10-4-2	30	135	108	132	107
T10-4-4	30	17178	49396	13706	41225
T20-4-2	30	57055	104187	49044	96469

Table 3: Average and standard deviation (SD) of the maximum number of pairs of a set for the cases where both methods solved the inference. Numbers are truncated.

7 Conclusion

We derived a new algorithm for exact posterior inference in extensively specified credal networks under strong independence. The algorithm is empirically shown to outperform an state-of-the-art method, being able to solve medium-sized networks in feasible time. We then showed that for networks of bounded treewidth and number of states per variable, a FPTAS for the problem exists. In our experiments, approximation and exact algorithms performed similar, likely due to the large constants hidden by the boundedness assumptions in the asymptotic complexity analysis.

Acknowledgements

This work was partially supported by the Swiss NSF grants n. 200020_134759 / 1, 200020-121785 / 1, and by the Hasler foundation grant n. 10030.

References

[1] C. Papadimitriou and M. Yannakakis. On the approximability of the trade-offs and optimal access of web sources. In *Proc. of the Annual Symp. on Foundations of Computer Science*, 2000.

[2] A. Antonucci and M. Zaffalon. Decision-Theoretic Specification of Credal Networks: A Unified Language for Uncertain Modeling with Sets of Bayesian Networks. *Intl. J. Approx. Reasoning* 49(2), 2008.

[3] A. Antonucci, Y. Sun, C. P. de Campos and M. Zaffalon. Generalized loopy 2U: a new algorithm for approximate inference in credal networks. *Intl. J. Approx. Reasoning* 55(5), 2010.

[4] D. Avis. Living with lrs. In *Discrete and Computational Geometry*, Lecture Notes in Computer Science, Springer, 2000.

[5] J. L. Bentley, H. T. Kung, M. Schkolnick and C. D. Thompson. On the average number of maxima in a set of vectors and applications. *J. of the ACM* 25, 1978.

[6] C. P. de Campos and F. G. Cozman. Inference in credal networks using multilinear programming. In *Proc. of the Starting AI Researchers' Symp.*, 2004.

[7] C. P. de Campos and F. G. Cozman. The inferential complexity of Bayesian and credal networks. In *Intl. Joint Conf. on Artif. Intelligence*, 2005.

[8] C. P. de Campos and F. G. Cozman. Inference in credal networks through integer programming. In *Proc. of the Intl. Symp. on Imprecise Probability: Theories and Applications*, 2007.

[9] A. Cano, M. Gomez, S. Moral and J. Abellan. Hill-climbing and branch-and-bound algorithms for exact and approximate inference in credal networks. In *Intl. J. Approx. Reasoning* 44(3), 2007.

[10] G. de Cooman, F. Hermans, A. Antonucci, and M. Zaffalon. Epistemic irrelevance in credal nets: The case of imprecise Markov trees. In *Intl. J. Approx. Reasoning* 51(9), 2010.

[11] F. G. Cozman. Credal networks. *Artif. Intelligence* 120(2), 2000.

[12] E. Fagiuoli and M. Zaffalon. 2U: an exact interval propagation algorithm for polytrees with binary variables. *Artif. Intelligence* 106(1), 1998.

[13] R. Haenni. Ordered valuation algebras: a generic framework for approximating inference. *Intl. J. Approx. Reasoning* 37(1), 2004.

[14] J. S. Ide , F. G. Cozman and F. T. Ramos. Generating Random Bayesian Networks with Constraints on Induced Width. In *Proc. of the European Conf. on Artif. Intelligence*, 2004.

[15] J. Kohlas. Information Algebras: Generic Structures for Inference. Springer-Verlag, 2003.

[16] J. C. F. da Rocha and F. G. Cozman. Inference in credal networks: branch-and-bound methods and the A/R+ algorithm. *Intl. J. Approx. Reasoning* 39(3), 2005.

[17] P. Shenoy and G. Shafer Axioms for Probability and Belief-Function Propagation. In *Proc. of the Conf. on Uncertainty in Artif. Intelligence*, 1988.

[18] P. Walley. Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, New York, 1991.