Generalized Loopy 2U: A New Algorithm for Approximate Inference in Credal Networks

Alessandro Antonucci, Marco Zaffalon, Yi Sun, Cassio P. de Campos
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Lugano, Switzerland

Abstract

Credal nets generalize Bayesian nets by relaxing the requirement of precision of probabilities. Credal nets are considerably more expressive than Bayesian nets, but this makes belief updating NP-hard even on polytrees. We develop a new efficient algorithm for approximate belief updating in credal nets. The algorithm is based on an important representation result we prove for general credal nets: that any credal net can be equivalently reformulated as a credal net with binary variables; moreover, the transformation, which is considerably more complex than in the Bayesian case, can be implemented in polynomial time. The equivalent binary credal net is updated by L2U, a loopy approximate algorithm for binary credal nets. Thus, we generalize L2U to non-binary credal nets, obtaining an accurate and scalable algorithm for the general case, which is approximate only because of its loopy nature. The accuracy of the inferences is evaluated by empirical tests.

1 Introduction

Bayesian nets (Sect. 2.1) are probabilistic graphical models based on precise assessments for the conditional probability mass functions of the net variables given the values of their parents. As a relaxation of such precise assessments, *credal nets* (Sect. 2.2) only require the conditional probability mass functions to belong to convex sets of mass functions, i.e., *credal sets*. Credal nets (CNs) are considerably more expressive than Bayesian nets,[1] and the price is an increased complexity of inference: *belief updating* in credal nets is NP-hard even on polytrees (de Campos and Cozman, 2005). The only known exception to this situation is the algorithm *2U* (Fagiuoli and Zaffalon, 1998), as it computes exact posterior beliefs on *binary* (i.e., with binary variables) polytree-shaped CNs in linear time. A *loopy* version of 2U (L2U) has been proposed for multiply connected binary credal nets by (Ide and Cozman, 2004). Inferences based on L2U are approximate, but a good accuracy is typically observed after few iterations (Sect. 2.3).

In this paper we develop an efficient algorithm for approximate belief updating of general CNs (any topology, number of states per variable). The algorithm is based on an important representation result that we prove in Appendix A: that any CN can be equivalently reformulated as one with binary variables. The corresponding transformation, which is considerably more complex than in the Bayesian case, is based on two distinct transformations: a *decision-theoretic specification* (Antonucci and Zaffalon, 2008), which augments the CN with control variables enumerating the multiple mass functions owned by the nodes of the net (Sect. 3.2); a *binarization* procedure (Antonucci et al., 2006) that transforms each variable into a cluster of binary variables (Sect. 3.1).

We prove that the sequential application of these two transformations, originally developed for independent reasons, returns an equiva-

[1] Greater expressiveness is a consequence of the fact that Bayesian nets are a subset of credal nets. Expressiveness should not be confused with informativeness: for example, it is thanks to the greater expressiveness that credal nets can model much less informative states of knowledge (including lack of knowledge) than those Bayesian nets can model.

lent binary representation of the original CN (Sect. 4.1). Such equivalent binary CN can be finally updated by L2U. Overall, that leads to a *generalized loopy 2U* (GL2U) algorithm for the updating in general CNs, whose only source of approximation is the loopy part (Sect. 4.2). The algorithm, which takes polynomial time (Sect. 4.3), has been implemented in a software. Experimental tests in Sect. 5 show that its accuracy is comparable to that of state-of-the-art approximate methods for CNs. This, together with its scalability, should make of GL2U the algorithm of choice especially for large nets.

2 Bayesian and Credal Nets

In this section we review the basics of Bayesian nets (BNs) and their extension to convex sets of probabilities, i.e., credal nets. Both the models are based on a collection of random variables, structured as a vector $\mathbf{X} := (X_1, \ldots, X_n)$,[2] and a directed acyclic graph (DAG) $\mathcal{G}$, whose nodes are associated with the variables of $\mathbf{X}$. In our assumptions the variables in $\mathbf{X}$ take values in finite sets. For both models, we assume the *Markov condition* to make $\mathcal{G}$ represent probabilistic independence relations between the variables in $\mathbf{X}$: every variable is independent of its non-descendant non-parents conditional on its parents. What makes BNs and CNs different is a different notion of independence and a different characterization of the conditional mass functions for each variable given the values of the parents, which will be detailed later.

Regarding notation, for each $X_i \in \mathbf{X}$, $\Omega_{X_i} = \{x_{i0}, x_{i1}, \ldots, x_{i(d_i-1)}\}$ denotes the possibility space of X_i, $P(X_i)$ is a mass function for X_i and $P(x_i)$ the probability that $X_i = x_i$, where x_i is a generic element of Ω_{X_i}. A similar notation with uppercase subscripts (e.g., X_E) denotes vectors (and sets) of variables in $\mathbf{X}$. Finally, the parents of X_i, according to $\mathcal{G}$, are denoted by Π_i, while for each $\pi_i \in \Omega_{\Pi_i}$, $P(X_i|\pi_i)$ is the mass function for X_i conditional on $\Pi_i = \pi_i$.

[2]The symbol ":=" is used for definitions.

2.1 Bayesian nets

For BNs, a conditional mass function $P(X_i|\pi_i)$ for each $X_i \in \mathbf{X}$ and $\pi_i \in \Omega_{\Pi_i}$ should be defined; and the standard notion of probabilistic independence is assumed in the Markov condition. A BN can therefore be regarded as a joint probability mass function over $\mathbf{X}$ that, according to the Markov condition, factorizes as follows:

$$P(\mathbf{x}) = \prod_{i=1}^{n} P(x_i|\pi_i), \tag{1}$$

for all the possible values of $\mathbf{x} \in \Omega_{\mathbf{X}}$, with the values of x_i and π_i consistent with $\mathbf{x}$. In the following, we represent a BN as a pair $\langle \mathcal{G}, P(\mathbf{X}) \rangle$. Posterior beliefs about a queried variable X_q, given evidence $X_E = x_E$, are defined as:

$$P(x_q|x_E) = \frac{\sum_{x_M} \prod_{i=1}^{n} P(x_i|\pi_i)}{\sum_{x_M, x_q} \prod_{i=1}^{n} P(x_i|\pi_i)}, \tag{2}$$

where $X_M := \mathbf{X} \setminus (\{X_q\} \cup X_E)$, the domains of the arguments of the sums are left implicit and the values of x_i and π_i are consistent with $\mathbf{x} = (x_q, x_M, x_E)$. Evaluating Eq. (2) is an NP-hard task, but for polytrees, Pearl's propagation allows for efficient updating (Pearl, 1988).

2.2 Credal sets and credal nets

CNs relax BNs by allowing for *imprecise probability* statements: in our assumptions, the conditional mass functions of a CN are required to belong to a *finitely generated* credal set, i.e., the convex hull of a finite number of mass functions for a certain variable. Geometrically, a credal set is a *polytope*. A credal set contains an infinite number of mass functions, but only a finite number of extreme mass functions corresponding to the *vertices* of the polytope. Updating based on a credal set is equivalent to that based only on its vertices (Walley, 1991). A credal set over X will be denoted as $K(X)$ and the set of its vertices as $\mathrm{ext}[K(X)]$. Given a non-empty $\Omega_X^* \subseteq \Omega_X$, an important credal set for our purposes is the *vacuous credal set* relative to Ω_X^*, i.e., the set of all the mass functions for X assigning probability one to Ω_X^*. We denote this set by $K_{\Omega_X^*}(X)$. In the following

we will use the well-known fact that the vertices of $K_{\Omega^*_X}(X)$ are the[3] $|\Omega^*_X|$ degenerate mass functions assigning probability one to the single elements of Ω^*_X. Marginalization generalizes to credal sets as follows: the marginalization $K(X)$ of a joint credal set $K(X,Y)$ to X is the convex hull of the mass functions $P(X)$ obtained from the marginalization of $P(X,Y)$ to X for each $P(X,Y) \in K(X,Y)$.

In order to specify a CN over the variables in $\mathbf{X}$ based on $\mathcal{G}$, a collection of conditional credal sets $K(X_i|\pi_i)$, one for each $\pi_i \in \Omega_{\Pi_i}$, should be provided separately for each $X_i \in \mathbf{X}$; while, regarding the Markov condition, we assume *strong independence* (Cozman, 2005). A CN associated with these local specifications is said to be with *separately specified* credal sets. Fig. 1 reports a CN, whose specification requires the (separate) assessment of an unconditional credal set for X_1, and two and eight conditional credal sets for X_2 and X_3. The specification becomes global considering the *strong extension* $K(\mathbf{X})$ of the CN, i.e., the convex hull of the following collection of joint mass functions:

$$\left\{ \prod_{i=1}^{n} P(X_i|\Pi_i) : P(X_i|\pi_i) \in K(X_i|\pi_i) \quad \begin{array}{l} \forall \pi_i \in \Omega_{\Pi_i}, \\ \forall i = 1,\ldots,n \end{array} \right\}. \tag{3}$$

We represent a CN as a pair $\langle \mathcal{G}, \mathbf{P}(\mathbf{X}) \rangle$, where $\mathbf{P}(\mathbf{X}) := \{P_k(\mathbf{X})\}_{k=1}^{n_v} := \text{ext}[K(\mathbf{X})]$. Clearly, for each $k = 1,\ldots,n_v$, $\langle \mathcal{G}, P_k(\mathbf{X}) \rangle$ is a BN. For this reason a CN can be regarded as a finite set of BNs. For CNs *updating* is intended as the computation of tight bounds of the posterior probabilities of a queried variable given some evidence, i.e., Eq. (2) generalizes as:

$$\underline{P}(x_q|x_E) = \min_{k=1,\ldots,n_v} \frac{\sum_{x_M} \prod_{i=1}^{n} P_k(x_i|\pi_i)}{\sum_{x_M,x_q} \prod_{i=1}^{n} P_k(x_i|\pi_i)}, \tag{4}$$

and similarly for upper probabilities $\overline{P}(x_q|x_E)$. Exact updating in CNs displays high complexity. Updating in polytree-shaped CNs is NP-complete, and NP^{PP}-complete in general CNs (de Campos and Cozman, 2005). The only known exact linear-time algorithm for updating a specific class of CNs is the 2U algorithm, which we review in the following section.

[3]The cardinality of a set Ω is denoted as $|\Omega|$.

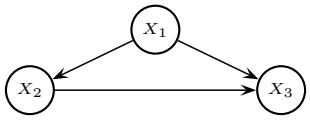

Figure 1: A separately specified CN over (X_1, X_2, X_3), with $|\Omega_{X_1}|=2$, $|\Omega_{X_2}|=|\Omega_{X_3}|=4$.

2.3 2U and its loopy extension

The adaptation of Pearl's updating algorithm to polytree-shaped binary CNs led to an exact algorithm called *2-Updating* (2U) (Fagiuoli and Zaffalon, 1998). Remarkably, 2U updates binary polytrees in time linear in the input size.

Loopy propagation is a popular technique that applies Pearl's propagation to multiply connected BNs: propagation is iterated until probabilities converge or for a fixed number of iterations. A loopy variant of 2U (L2U) can update multiply connected binary CNs in an approximate way (Ide and Cozman, 2004). Initialization of variables and messages follows the same steps used in the 2U algorithm. Then nodes are repeatedly updated, until convergence of probabilities is observed.[4] L2U is basically an iteration of 2U and its complexity is therefore linear in the input size and in the number of iterations. Overall, the L2U algorithm is an excellent algorithm, it is fast and returns good results with low errors after a few iterations.

3 Transformations of Credal Nets

In this section we review two different transformations of CNs that have been recently proposed for independent reasons. Their sequential application is the basis to obtain an equivalent representation of CNs based on binary variables.

3.1 Binarization algorithm

By definition, L2U (see Sect. 2.3) cannot be applied to a non-binary CN in the example of Fig. 1. To overcome this limitation, a *binarization* that transforms a CN into a binary CN has been proposed in (Antonucci et al., 2006).

First, each variable is equivalently represented by a cluster of binary variables. Assume

[4]Despite the lack of a formal proof, convergence of L2U has been always observed in the numerical tests.

d_i, which is the number of states for X_i, to be an integer power of two, and let $\tilde{d}_i := \log_2 |\Omega_{X_i}|$.[5] An obvious one-to-one correspondence between the states of X_i and the joint states of a vector of $\tilde{d}_i$ binary variables $\tilde{X}_i := (\tilde{X}_i^0, \tilde{X}_i^1, \ldots, \tilde{X}_i^{\tilde{d}_i-1})$ is established if the joint state $(\tilde{x}_i^0, \ldots, \tilde{x}_i^{\tilde{d}_i-1}) \in \{0,1\}^{\tilde{d}_i}$ is associated with $x_{il} \in \Omega_{X_i}$, where l is the integer whose $\tilde{d}_i$-bit representation is $\tilde{x}_i^{\tilde{d}_i-1} \cdots \tilde{x}_i^1 \tilde{x}_i^0$. Elements of $\tilde{X}_i$ are said *bits* of X_i and their position in the vector their *order*.

Overall, $\tilde{\mathbf{X}}$ denotes the vector of bits obtained *binarizing* all the elements of $\mathbf{X}$. We write $P(\mathbf{X}) = \tilde{P}(\tilde{\mathbf{X}})$, if $P(\mathbf{x}) = \tilde{P}(\tilde{\mathbf{x}})$ for each $\mathbf{x} \in \Omega_{\mathbf{X}}$, where $\tilde{\mathbf{x}} \in \Omega_{\tilde{\mathbf{X}}}$ is the state corresponding to $\mathbf{x}$.

A DAG $\tilde{\mathcal{G}}$ associated to the variables $\tilde{\mathbf{X}}$ can be obtained from $\mathcal{G}$ as follows: (i) two nodes of $\tilde{\mathcal{G}}$ corresponding to bits of different variables in $\mathbf{X}$ are connected by an arc if and only if there is an arc with the same direction between the related variables in $\mathbf{X}$; (ii) an arc connects two nodes of $\tilde{\mathcal{G}}$ corresponding to bits of the same variable of $\mathbf{X}$ if and only if the order of the bit associated to the node from which the arc departs is lower than the order of the bit associated to the remaining node. An example of this transformation is depicted in Fig. 2.

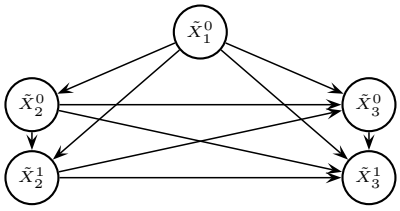

Figure 2: The binarization of the CN in Fig. 1.

Finally, regarding the quantification of the conditional credal sets, we have:

$$\underline{\tilde{P}}(\tilde{x}_i^j|\tilde{\pi}_i^j) := \min_{k=1,\ldots,n_v} \tilde{P}_k(\tilde{x}_i^j|\tilde{\pi}_i^j), \qquad (5)$$

where the index k is defined like in Eq. (4). Denoting by $\tilde{\Pi}_i$ the parents of $\tilde{X}_i^j$ corresponding to the binarization of Π_i, i.e., those that are not in the same cluster of $\tilde{X}_i^j$, the probabilities to be minimized on the right-hand side are:

$$\tilde{P}_k(\tilde{x}_i^j|\tilde{x}_i^{j-1}, \ldots, \tilde{x}_i^0, \tilde{\pi}_i) \propto \sum\nolimits_l^* P_k(x_{il}|\pi_i), \qquad (6)$$

where the sum $\sum^*$ is restricted to the states $x_{il} \in \Omega_{X_i}$ such that $l \bmod 2^{j+1}$ is the integer whose $(j+1)$-bit representation is $\tilde{x}_i^j, \ldots, \tilde{x}_i^1, \tilde{x}_i^0$, π_i is the joint state of the parents of X_i corresponding to the joint state $\tilde{\pi}_i$ for the bits of the parents of X_i, symbol $\propto$ denotes proportionality, and the relations are considered for each $i = 1, \ldots, n$, $j = 0, \ldots, \tilde{d}_i - 1$, and $\pi_i \in \Omega_{\Pi_i}$.

If both the states of $\tilde{X}_i^j$ produce zero in Eq. (6), the corresponding conditional mass functions can be arbitrarily specified (we set a degenerate mass function). Note that minimization in Eq. (5) can be obtained by simply considering the vertices of $K(X_i|\pi_i)$ in Eq. (6).

The overall procedure returns a well-defined CN, which is called the *binarization* of the original CN. Given an updating problem on a CN as in Eq. (4), we can consider the corresponding problem on its binarization. E.g., the computation of $\underline{P}(x_{33}|x_{10})$ for the CN in Fig. 1 corresponds to $\underline{\tilde{P}}(\tilde{X}_3^0 = 1, \tilde{X}_3^1 = 1|\tilde{X}_1^0 = 0)$. According to (Antonucci et al., 2006, Th. 2) this is an *outer approximation* (i.e., the posterior interval includes that of the original updating problem), which can be approximately estimated by L2U.

This approach entails a twofold approximation: (i) the approximation introduced by the binarization and (ii) that due to the loopy propagation. Approximation (i) can be regarded as originated by replacing each credal set of the original net with an enclosing polytope that can have a smaller number of vertices. By construction, the latter number cannot be controlled and could be too low to lead to a satisfactory approximation of the original credal set, which in turns leads approximation (i) to be quite crude. In the next section, we recall an independently developed transformation that will be used to remove approximation (i).

3.2 Decision-theoretic specification

In (Antonucci and Zaffalon, 2008), a general graphical language for CNs based on the so-called *decision-theoretic specification* (DTS) has

[5] This is not a limitation as a number of *dummy* states up the the nearest power of two can be always added. From now on we assume for all the variables a number of possible values equal to an integer power of two.

been proposed. A DTS of a CN is obtained augmenting the original CN by a number of *control nodes*, used to enumerate the vertices of the conditional credal sets. That turns the original nodes into precise-probability ones, while the control nodes can be formulated as chance nodes with vacuous credal sets.

Let us briefly describe this transformation in the case of a CN $\langle \mathcal{G}, \mathbf{P}(\mathbf{X}) \rangle$. First, we obtain from $\mathcal{G}$ a second DAG $\mathcal{G}'$ defined over a wider domain $\mathbf{X}' := (X_1, \ldots, X_{2n})$. This is done by iterating, for each $i = 1, \ldots, n$, the following operations: (i) add a node X_{i+n}; (ii) draw an arc from each parent of X_i to X_{i+n}; (iii) delete the arcs connecting the parents of X_i with X_i; (iv) draw an arc from X_{i+n} to X_i. An example of this transformation is shown in Fig. 3.

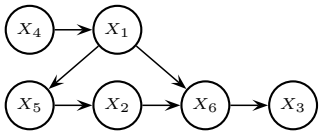

Figure 3: The output of the transformation described in Sect. 3.2 for the CN in Fig. 1.

Note that, for each $i = 1, \ldots, n$, $\Pi'_{i+n} = \Pi_i$, i.e., the parents of X_{i+n} in $\mathcal{G}'$ are the parents of X_i in $\mathcal{G}$, and also $\Pi'_i = X_{i+n}$, i.e., X_{i+n} is the only parent of X_i in $\mathcal{G}'$ and is therefore called the *control variable* of X_i.

We assume a one-to-one correspondence between the possible states of a control variable X_{i+n} and the collection of all the (distinct) extreme mass functions of *all* the conditional credal sets specified over X_i, i.e., $\Omega_{X_{i+n}} := \bigcup_{\pi_i \in \Omega_{\Pi_i}} \text{ext}[K(X_i|\pi_i)]$, for each $i = 1, \ldots, n$. As an example, assuming the number of vertices for the credal sets of the CN in Fig. 1 equal to the number of possible states of the relative variables, we have that X_4 in Fig. 3 is a binary variable, whose states correspond to the two vertices of $K(X_1)$; X_5 has eight possible states corresponding to the four vertices of $K(X_2|x_1)$ and the four of $K(X_2|\neg x_1)$; X_6 has 32 possible states corresponding to the vertices, four per each set, of the conditional credal sets over X_3.

Finally, in order to obtain a well-defined CN over $\mathbf{X}'$ associated to $\mathcal{G}'$, we quantify the conditional credal sets as follows. For each $i = 1, \ldots, n$, we set $K'(X_i|x_{i+n}) := P(X_i)_{x_{i+n}}$, where $P(X_i)_{x_{i+n}}$ is the element of $\text{ext}[K(X_i|\pi_i)]$ corresponding to x_{i+n}. For the control nodes $\{X_{i+n}\}_{i=1}^n$, we set $K'(X_{i+n}|\pi'_{i+n}) := K_{\Omega^{\pi_i}_{X_{i+n}}}(X_i)$, where $\Omega^{\pi_i}_{X_{i+n}} \subseteq \Omega_{X_{i+n}}$ is the set of vertices of $K(X_i|\pi_i)$.

The CN returned by this transformation will be denoted as $\langle \mathcal{G}', \mathbf{P}'(\mathbf{X}') \rangle$, and its strong extension as $K'(\mathbf{X}')$. Remarkably, $\langle \mathcal{G}', \mathbf{P}'(\mathbf{X}') \rangle$ provides an equivalent representation of $\langle \mathcal{G}, \mathbf{P}(\mathbf{X}) \rangle$ being that $K'(\mathbf{X}) = K(\mathbf{X})$ as stated by Th. 2 in (Antonucci and Zaffalon, 2008), where $K'(\mathbf{X})$ is the marginalization of $K'(\mathbf{X}')$ to $\mathbf{X}$.

4 Exact Binarization & GL2U

Now we present the original contributions of this paper, consisting of a general representation result (Sect. 4.1), the definition of GL2U (Sect. 4.2), the study of its computational complexity, and its empirical evaluation (Sect. 5).

4.1 Exact binarization

Consider the sequential application of the transformations detailed in Sect. 3.2 and Sect. 3.1. Thus, given a CN $\langle \mathcal{G}, \mathbf{P}(\mathbf{X}) \rangle$, obtain $\langle \mathcal{G}', \mathbf{P}'(\mathbf{X}') \rangle$ by a DTS, and hence $\langle \tilde{\mathcal{G}}', \tilde{\mathbf{P}}'(\tilde{\mathbf{X}}') \rangle$ through binarization. The latter CN is said the *exact binarization* of the first, a terminology justified by the following result.

Theorem 1. *Consider a CN $\langle \mathcal{G}, \mathbf{P}(\mathbf{X}) \rangle$ and its exact binarization $\langle \tilde{\mathcal{G}}', \tilde{\mathbf{P}}'(\tilde{\mathbf{X}}') \rangle$. Let $K(\mathbf{X})$ and $\tilde{K}'(\tilde{\mathbf{X}}')$ be their corresponding strong extensions. Then:*

$$K(\mathbf{X}) = \tilde{K}'(\tilde{\mathbf{X}}), \tag{7}$$

with $\tilde{K}'(\tilde{\mathbf{X}})$ marginalization of $\tilde{K}'(\tilde{\mathbf{X}}')$ to $\tilde{\mathbf{X}}$.

According to Eq. (7), $\langle \tilde{\mathcal{G}}', \tilde{\mathbf{P}}'(\tilde{\mathbf{X}}') \rangle$ is an equivalent binary representation of $\langle \mathcal{G}, \mathbf{P}(\mathbf{X}) \rangle$. It should be pointed out that, even if we focus on the case of CNs with separately specified credal sets, Th. 1 holds also for so-called *non-separately specified* CNs, for which a DTS can be provided as well. Similarly, the algorithm presented in the next section can be applied to any CN, separately or non-separately specified.

4.2 GL2U

Th. 1 is a basis for the solution of general inference problems, as stated by the following straightforward corollary.

Corollary 1. *Any inference problem on a CN can be equivalently computed in its exact binarization.*

According to Cor. 1, we can consider a so-called *generalized* L2U algorithm (GL2U), where given an updating problem on a CN, we solve by L2U the corresponding updating problem on the exact binarization of the original CN. The overall procedure is still approximate, but differently from the case without DTS considered in (Antonucci et al., 2006), the only source of approximation is the loopy component.

4.3 Complexity issues

According to the discussion in the previous section, the computational time required by GL2U to update a CN $\langle \mathcal{G}, \mathbf{P}(\mathbf{X}) \rangle$ is basically that required by L2U to update $\langle \tilde{\mathcal{G}}', \tilde{\mathbf{P}}'(\tilde{\mathbf{X}}') \rangle$. This is $O(t \cdot 2^{2a})$, where t is the number of iterations and a the maximum indegree of $\mathcal{G}'$. It can be checked that $\tilde{X}_i^{\tilde{d}_i - 1}$ has the maximum indegree among the $\tilde{d}_i$ binary nodes in the cluster $\tilde{X}_i$; similarly, $\tilde{X}_{i+n}^{\tilde{d}_{i+n}-1}$ has the maximum indegree among the $\tilde{d}_{i+n}$ nodes of $\tilde{X}_{i+n}$. Note also that the number of nodes in $\tilde{\Pi}_i$ is $\sum_{j/X_j \in \Pi_i} \tilde{d}_j$. Therefore, the indegrees of $\tilde{X}_i^{\tilde{d}_i - 1}$ and $\tilde{X}_{i+n}^{\tilde{d}_{i+n}-1}$ are respectively $\tilde{d}_i + \tilde{d}_{i+n} - 1$ and $\tilde{d}_{i+n} + \sum_{j/X_j \in \Pi_i} \tilde{d}_j - 1$. Thus, considering that by definition $2^{\tilde{d}_i} = d_i$,[6] the local complexity of the algorithm for these two worst cases is respectively $O(t \cdot (d_i \cdot d_{i+n})^2)$ and $O(t \cdot (d_{i+n} \cdot \prod_{j/X_j \in \Pi_i} d_j)^2)$.

Globally, any iteration of 2U is linear in the size (i.e., the longest path) of the net, and the size of the exact binarization grows of a factor at most equal to $2 \cdot \max_{i=1}^{2n} \tilde{d}_i$ with respect to the original net. The factor depends (i) on the decision-theoretic transformation that doubles the number of nodes, and on (ii) the binarization that makes of each node $X_i \in \mathbf{X}'$ a cluster of binary nodes $\tilde{X}_i$ whose size depends on the logarithm $\tilde{d}_i$ of its number of states d_i. We can approximate the global complexity by assuming that a not-too-big constant bounds both the logarithms of (i) the maximum number of states for each variable in $\mathbf{X}$, and (ii) the maximum overall number of vertices of the credal sets associated to these variables. Thus, we conclude that any iteration of GL2U is roughly linear in the size of the net.

[6] As in Sect. 4.1, the number of states for each variable in $\mathbf{X}'$ is assumed to be an integer power of two. The discussion of the general case is omitted because of lack of space and will be presented in a future work.

5 Numerical Tests

In order to test the performance of GL2U, we have chosen the *Alarm* and the *Insurance* nets, as well as some random generated nets. We work with random polytrees with 50 nodes (Polyt-50), and random multiply connected nets with 10 and 25 nodes (Multi-10 and Multi-25, respectively). For the Alarm and the Insurance nets, we use the original graph with the original number of states. For the random polytrees, we generate random graphs with 50 nodes and at most 4 categories in each variable. With random multiply connected nets, we work with 10 and 25 nodes, and 4 and 8 categories.

Table 1: Average mean square errors of LS, GL2U and BIN. Maximum number of states/vertices is indicated in the second column. Best accuracies are bolded.

		LS	GL2U	BIN
Multi-10	4 / 2	0.0189	**0.0140**	0.0181
Multi-10	8 / 2	0.0195	**0.0107**	0.0338
Multi-10	4 / 4	**0.0120**	0.0175	0.0308
Multi-10	4 / 8	**0.0027**	0.0125	0.0222
Multi-10	8 / 4	0.0234	**0.0189**	0.0693
Multi-25	4 / 2	0.0231	**0.0160**	0.0184
Multi-25	4 / 4	0.0248	**0.0204**	0.0303
Polyt-50	4 / 2	**0.0112**	0.0193	0.0289
Polyt-50	4 / 4	**0.0145**	0.0221	0.0392
Insurance	5 / 2	**0.0055**	0.0117	0.0175
Insurance	5 / 4	**0.0113**	0.0132	0.0193
Alarm	4 / 2	0.0290	**0.0190**	0.0302
Alarm	4 / 4	0.0331	**0.0239**	0.0423

We run marginal inferences using GL2U, the "rough" binarization without DTS (BIN), the approximate *local search* method (da Rocha et al., 2003) (LS) limited to 20 iterations in order to have running times similar to those of GL2U, and the exact method presented in (de Campos and Cozman, 2007). Tab. 1 shows

the mean square errors. GL2U improves, often substantially, the approximation accuracy when compared to BIN; moreover, it has accuracy similar to LS. Moreover, the running time and the amount of allocated memory for LS rapidly increases with the size of the net, that makes unfeasible a solution for large nets, which can be instead quickly updated by GL2U (see Fig. 4).[7] As far as we know other existing algorithms besides LS are at least exponential in the treewidth of the moralized graph and suffer from the same complexity issues. In fact, some comparisons have been done also with the *hill-climbing* algorithm in (Cano et al., 2007) and a behavior very similar to that in Fig. 4 has been observed. Hence, GL2U has a great expected speed up with respect to them.

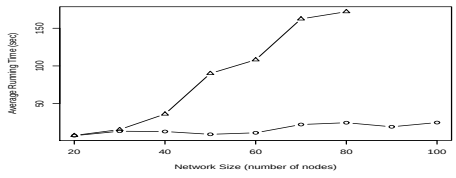

Figure 4: Average running time versus net size for LS (triangles) and GL2U (circles). LS cannot solve CNs with more than 80 nodes.

6 Conclusions

This paper has proposed a new approximate algorithm for CNs updating. This task is achieved augmenting the net by a number of nodes enumerating the vertices of the credal sets and then transforming the CN in a corresponding net over binary variables, and updating such binary CN by the loopy version of 2U. The procedure applies to any CN, without restrictions related to the topology or the number of possible states, and the only approximation is due to the loopy propagation. Empirical analysis shows that GL2U is a competitive procedure for approximate inference in CNs both in terms of accuracy and scalability. The algorithm is purely distributed and allows for simultaneous updating of all the variables in the net: these characteristics are usually not shared by optimization-based algorithms. Moreover, the computational complexity of GL2U makes it possible to solve large nets, which cannot be updated, at least with the same accuracy, by existing algorithms.

[7] A software implementation of GL2U is freely available at www.idsia.ch/∼sun/g2lu.html. The running times in Fig. 4 refer to an earlier implementation, while the last release is significantly faster.

Acknowledgments

Work partially supported by the Swiss NSF grants 200021-113820/1 and 200020-116674/1, and Hasler Foundation grant 2233.

A Proofs

Lemma 1. *Consider a CN with a single node X and vacuous $K(X) := K_{\Omega^*_X}(X)$, where $\Omega^*_X \subseteq \Omega_X$. Let $\tilde{K}(\tilde{X})$ denote the strong extension of its binarization (as in Sect. 3.1). Then:*

$$\tilde{K}(\tilde{X}) = K(X). \tag{8}$$

Proof. Consider a generic $\tilde{P}_*(\tilde{X}) \in \text{ext}[\tilde{K}(\tilde{X})]$, where $\tilde{X} := (\tilde{X}^0, \ldots, \tilde{X}^{\tilde{d}-1})$ with $\tilde{d}$:=$\log_2|\Omega_X|$. A corresponding mass function $P_*(X) := \tilde{P}_*(\tilde{X})$ can be therefore defined. Thus:

$$\tilde{P}_*(\tilde{x}) = \prod_{j=0}^{\tilde{d}-1} \tilde{P}_*(\tilde{x}^j|\tilde{x}^{j-1}, \ldots, \tilde{x}^0), \tag{9}$$

for each $\tilde{x} \in \Omega_{\tilde{X}}$ such that $(\tilde{x}^0, \ldots, \tilde{x}^{\tilde{d}-1}) = \tilde{x}$. For each j=$0, \ldots, \tilde{d}-1$ and each possible value of their parents, the conditional mass functions $\tilde{P}_*(\tilde{X}^j|\tilde{x}^{j-1}, \ldots, \tilde{x}^0)$ are vertices of their corresponding conditional credal sets because of Proposition 1 of (Antonucci and Zaffalon, 2008). Thus, the values of the conditional probabilities on the right-hand side of Eq. (9) are obtained by a minimization as in Eq. (5). The values to be minimized are obtained from Eq. (6), where the conditional probabilities on the right-hand side are the vertices of $K(X)$, i.e., the $m := |\Omega^*_X|$ degenerate extreme mass functions of the vacuous credal set $K_{\Omega^*_X}(X)$. This means that there is only a non-zero term in the sum in Eq. (6) and therefore each vertex of $K_{\Omega^*_X}$ produces a degenerate conditional mass function for the corresponding binary variable. Consequently, also the extreme values returned by Eq. (5) will be

degenerate. We can therefore conclude that, according to Eq. (9), also $\tilde{P}_*(\tilde{X})$ and hence $P_*(X)$ is a degenerate mass functions. Let $x_* \in \Omega_X$ be the state of X such that $P_*(x_*) = 1$. Considering Eq. (9) for $\tilde{x}_* \in \Omega_{\tilde{X}}$, we conclude that all the conditional probabilities on the right-hand side are equal to one. Considering the highest order bit, according to Eq. (6) and denoting by $P_k(X)$ a vertex of $\Omega_*(X)$, we have $\tilde{P}_*(\tilde{x}_*^{\tilde{d}-1}|\tilde{x}_*^{\tilde{d}-2}, \ldots, \tilde{x}_*^0) = P_k(x_*) = 1$, that requires $x_* \in \Omega_X^*$. Thus, $P_*(X) \in \text{ext}[K(X)]$, that implies $\text{ext}[\tilde{K}(\tilde{X})] \subseteq \text{ext}[K(X)]$, and finally $\tilde{K}(\tilde{X}) \subseteq K(X)$. On the other side, $\tilde{K}(\tilde{X}) \supseteq K(X)$ because of Th. 2 in (Antonucci et al., 2006), and hence the thesis. □

Proof of Th. 1. Given a $\tilde{P}'_*(\tilde{\mathbf{X}}') \in \text{ext}[\tilde{K}'(\tilde{\mathbf{X}}')]$, the following factorization holds:

$$\tilde{P}'_*(\tilde{\mathbf{x}}') = \prod_{i=1}^{2n} \prod_{j=0}^{\tilde{d}_i - 1} \tilde{P}'_*(\tilde{x}_i^j|\tilde{\pi}_i^j) = \prod_{i=1}^{2n} \tilde{P}'_*(\tilde{x}_i^0, \ldots, \tilde{x}_i^{\tilde{d}_i-1}|\tilde{\pi}'_i), \tag{10}$$

for each $\tilde{\mathbf{x}}' \in \Omega_{\tilde{\mathbf{X}}'}$, where the values of the other variables are consistent with $\tilde{\mathbf{x}}$, and the last equality follows from chain rule. Eq. (10) defines $P'_*(X_i|\pi'_i) := \tilde{P}'_*(\tilde{X}_i^0, \ldots, \tilde{X}_i^{\tilde{d}_i-1}|\tilde{\pi}'_i)$. As noted in Sect. (3.2), for each $i = 1, \ldots, n$ and $\pi_i \in \Omega_{\Pi_i}$, $K'(X_i|\pi'_i)$ is a credal set made of a single point. Thus, as a corollary of Th. 1 in (Antonucci et al., 2006), we have $P'_*(X_i|\pi'_i) \in \text{ext}[K'(X_i|\pi'_i)]$, being in fact the only element of this credal set. Similarly, for each $i = 1, \ldots, n$, the credal set $K'(X_{i+n}|\pi'_{i+n})$ is vacuous. Thus, regarding this credal set as a CN made of a single node, we invoke Lemma 1 and obtain from $\tilde{P}'_*(\tilde{X}_{i+n}|\tilde{\pi}'_{i+n}) \in \text{ext}[\tilde{K}'(\tilde{X}_{i+n}|\tilde{\pi}'_{i+n})]$ that $P'_*(X_{i+n}|\pi'_{i+n}) \in \text{ext}[K'(X_{i+n}|\pi'_{i+n})]$. Overall, we proved that $P'_*(\mathbf{X}')$ is a combination of local vertices of the credal sets of $\langle \mathcal{G}', \mathbf{P}'(\mathbf{X}')\rangle$. Thus, $P'_*(\mathbf{X}') \in \text{ext}[K'(\mathbf{X}')]$, from which $\text{ext}[\tilde{K}'(\tilde{\mathbf{X}}')] \subseteq \text{ext}[K'(\mathbf{X}')]$, and finally $\tilde{K}'(\tilde{\mathbf{X}}') \subseteq K'(\mathbf{X}')$. According to Lemma 1 in (Antonucci et al., 2006), $\tilde{K}'(\tilde{\mathbf{X}}') \supseteq K'(\mathbf{X}')$. Thus, $\tilde{K}'(\tilde{\mathbf{X}}') = K'(\mathbf{X}')$. Marginalizing on both the sides we get $\tilde{K}'(\tilde{\mathbf{X}}) = K'(\mathbf{X})$. But Th. 2 in (Antonucci and Zaffalon, 2008) states $K(\mathbf{X}) = K'(\mathbf{X})$, from which the thesis. □

References

A. Antonucci and M. Zaffalon. 2008. Decision-theoretic specification of credal networks: A unified language for uncertain modeling with sets of Bayesian networks. *Int. J. Approx. Reasoning*. Forthcoming.

A. Antonucci, M. Zaffalon, J. S. Ide, and F. G. Cozman. 2006. Binarization algorithms for approximate updating in credal nets. In *Proceedings of the third European Starting AI Researcher Symposium*, pages 120–131, Amsterdam. IOS Press.

A. Cano, M. Gómez, S. Moral, and J. Abellán. 2007. Hill-climbing and branch-and-bound algorithms for exact and approximate inference in credal networks. *International Journal of Approximate Reasoning*, 44(3):261–280.

F. G. Cozman. 2005. Graphical models for imprecise probabilities. *Int. J. Approx. Reasoning*, 39(2–3):167–184.

J. C. da Rocha, F. G. Cozman, and C. P. de Campos. 2003. Inference in polytrees with sets of probabilities. In *Conference on Uncertainty in Artificial Intelligence*, pages 217–224, Acapulco.

C. P. de Campos and F. G. Cozman. 2005. The inferential complexity of Bayesian and credal networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1313–1318, Edinburgh.

C. P. de Campos and F. G. Cozman. 2007. Inference in credal networks through integer programming. In *Proceedings of the Fifth International Symposium on Imprecise Probability: Theories andApplications*, Prague. Action M Agency.

E. Fagiuoli and M. Zaffalon. 1998. 2U: an exact interval propagation algorithm for polytrees with binary variables. *Artificial Intelligence*, 106(1):77–107.

J. S. Ide and F. G. Cozman. 2004. IPE and L2U: Approximate algorithms for credal networks. In *Proceedings of the Second Starting AI Researcher Symposium*, pages 118–127, Amsterdam. IOS Press.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo.

P. Walley. 1991. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, New York.