IDS: A Divide-and-Conquer Algorithm for Inference in Polytree-Shaped Credal Networks

José Carlos F. da Rocha[1]*, **Cassio P. de Campos**[2] , **Fabio G. Cozman**[3]†

[1]Universidade Estadual de Ponta Grossa
Av. Carlos Cavalcanti, 4748 - Campus de Uvaranas
84.030-900 - Ponta Grossa - PR

[2]Pontifícia Universidade Católica de São Paulo
São Paulo - SP

[3]Escola Politécnica da Universidade de São Paulo
São Paulo - SP

`jrocha@uepg.br,cassio@pucsp.br,fgcozman@usp.br`

Abstract. *A credal network is a graph-theoretic model that represents imprecision in joint probability distributions. An inference in a credal net aims at computing an interval for the probability of an interest event. The algorithms for inference in credal networks can be divided into exact and approximate. The selection of such an algorithm is based on a trade off that ponders how much time someone wants to spend in a particular calculation against the quality of the computed values. This paper presents an algorithm, called IDS, that combines exact and approximate methods for computing inferences in polytree-shaped credal networks. The algorithm provides an approach to trade time and precision when making inferences in credal nets.*

Resumo. *Uma rede credal é um formalismo baseado em grafos que representa imprecisão em distribuições conjuntas. Uma inferência em uma rede credal objetiva o cômputo de um intervalo de probabilidades para um evento de interesse. Os algoritmos para inferência em redes credais podem ser classificados como exatos ou aproximados. A seleção de um algoritmo exige uma análise de custo×benefício que pondera quanto tempo se deseja gastar no cálculo de um intervalo em relação a qualidade das aproximações. Este artigo apresenta um algoritmo, chamado IDS, que combina métodos exatos e aproximados no cômputo de inferências em redes com topologia em polytree e que provê uma estratégia para limitar o esforço computacional empregado em uma inferência.*

1. Introduction

The formalism of *Bayesian networks* offers a graph-theoretic model that compactly encodes joint distributions [Pearl, 1988]. In a Bayesian network, every probability value must be exact. This constraint is useful computationally as it leads to efficient algorithms for computation of marginal and conditional probabilities. However, sometimes practical and theoretical difficulties are in the way of exact probability specification [Walley, 1991]; it has motivated the development of several approaches to represent probabilistic imprecision in Bayesian networks [Fertig and Breese, 1990, Ha et al., 1998, Wellman, 1990]. In

*Supported by UEPG and CAPES.

†Partially support by a grant from CNPq. This work was (partially) developed with support from HP Brazil R&D

this work we consider the formalism of *credal networks* [Fagiouli and Zaffalon, 1998, Cozman, 2000]. More particularly, we are interested in *extensions* of credal networks that encode a specific type of independence relation called *strong independence* [Couso et al., 1999]. An *inference* in a credal network is a computation that produces a lower/upper probability for an interest event. Recently, efficient inference algorithms have been proposed that can handle credal networks containing dozens of variables [Campos and Cozman, 2004]. However, it is still impossible to obtain exact inferences in large networks. Naturally one is led to consider algorithms for approximate inference [Cano and Moral, 1999, Rocha et al., 2003, Tessem, 1992].

In this paper we present a new algorithm called IDS (Inference by Decomposition in Subnetworks) that combines approximate and exact strategies to compute outer bounds for the extreme values of a probability interval in polytree-shaped credal networks. The idea of combining different inference algorithms is not new in Bayesian networks but is has not been explored in connection with credal networks. The central idea of IDS is rather simple: divide a network in parts, and run exact and approximate algorithms on different parts of the network, looking for a trade-off between time and quality of results.

The article is organized as follows. Section 2 presents a brief review of credal networks and inference algorithms. Section 3 describes the IDS algorithm, and Section 4 discusses an example. Section 5 presents our final comments.

2. Background

A *credal set* $K(X)$ for random variable X is a set of probability distributions for X [Levi, 1980]. In this paper we deal only with categorical variables, and we consider only credal sets that can be represented by the convex hull of finitely many distributions. Credal sets may contain either joint, conditional or marginal distributions. Given a set of random variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, a joint credal set $K(\mathbf{X})$ contains joint distributions $p(\mathbf{X})$. For any variable $X_i \in \mathbf{X}$, the marginal credal set $K(X_i)$ can be computed by marginalizing each extreme distribution of $K(\mathbf{X})$ and by taking the convex hull of the marginalized distributions. Similarly, given the event $\{X_i = x_{i,e}\}$, for $X_i \in \mathbf{X}$, it is possible to calculate a conditional credal set $K(\mathbf{X} \setminus \{X_i\}|x_{i,e}) = \{p_1(\mathbf{X} \setminus \{X_i\}|x_{i,e}), \ldots, p_t(\mathbf{X} \setminus \{X_i\}|x_{i,e})\}$ from $K(\mathbf{X})$. We only need to condition[1] every extreme point of $K(\mathbf{X})$ on $\{X_i = x_{i,e}\}$ and take the convex hull of the previously calculated distributions.

Let $\mathbf{Y}$ and $\mathbf{Z}$ be two proper and disjoint subsets of $\mathbf{X}$. The conditional information of $\mathbf{Y}$ given $\mathbf{Z}$ can be represented in many ways [Moral, 1999]. Here we assume that this information is given as collection of *separately specified* credal sets $\mathbf{Q}(\mathbf{Y}|\mathbf{Z}) = \{K(\mathbf{Y}|z_0), \ldots, K(\mathbf{Y}|z_t)\}$. Note that we have a collection of conditional credal sets; there is a credal set defined on $\mathbf{Y}$ for every joint event $z_k \in \mathbf{Z}$, and the constraints that define these credal sets have no relation to each other.

Given a marginal credal set $K(X_q)$ it is possible to compute the probability interval of any category $x_{q,i}$ of X_q just by computing the minimum and the maximum probability of $\{X_q = x_{q,i}\}$ over all distributions in the credal set. These values are called *lower* and *upper* probabilities and are defined as $\underline{P}(X_q = x_{q,i}) = \min_{p(X_q) \in K(X_q)} P(X_q = x_{q,i})$ and $\overline{P}(X_q = x_{q,i}) = \max_{p(X_q) \in K(X_q)} P(X_q = x_{q,i})$, respectively. Such interval results can be useful for classification [Zaffalon, 1998] and robustness analysis [Cozman, 1997].

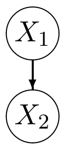
If we have a large number of variables to handle, it may be hard to represent a joint credal set. First, it is hard to represent each one of the joint distributions over a large

[1] We assume throughout that any conditioning even has lower probability larger than zero.

number of variables. Second, the number of extreme distributions in a joint credal set may be huge. Credal networks offer a compact representation that can mitigate some of these hurdles. Let $\mathbf{X}$ be the set of random variables; we take a credal network $\mathbf{C}$ over $\mathbf{X}$ to consist of:

- a directed acyclic graph $\mathbf{G}$ in which every node represents a single random variable in $\mathbf{X}$ and the every arc represents a direct dependency between variables, and where $\rho(X_i)$ and $\chi(X_i)$ denote respectively the parents and children of X_i in $\mathbf{G}$.
- each node X_i is associated with a collection of separately specified credal sets $\mathbf{Q}(X_i|\rho(X_i))$.

The Figure 1 shows a simple network with two binary variables, X_1 and X_2. This network has two collections of separately specified credal sets: one collection, $\mathbf{Q}(X_1)$, contains just one marginal credal; the other collection, $\mathbf{Q}(X_2|X_1)$, contains conditional credal sets.



$\mathbf{Q}(X_1) = \{K(X_1) = cc((0.5; 0.5), (0.3; 0.7))\ \}$
$\mathbf{Q}(X_2|X_1) = \{K(X_2|x_{1,1}), K(X_2|x_{1,2})\}$
with
$K(X_2|x_{1,1}) = cc((0.5; 0.5), (0.3; 0.7))$
$K(X_2|x_{1,2}) = cc((0.4; 0.6), (0.2; 0.8))$

Figure 1: A simple credal net.

We assume that a credal network satisfies the following *Markov condition*: every variable is *strongly* independent of its nondescendants nonparents given its parents. Note that we adopt the concept of *strong independence*. Two variables X_a and X_b are strong independent if for every vertice of their joint credal set we have $P(X_a = x_{a,i} X_b = x_{b,j}) = P(X_a = x_{a,i}) \cdot P(X_b = x_{b,j})$, for all events of X_a and X_b. Two variables X_a and X_b are strong independent conditional on a variable X_c when $P(X_a = x_{a,i}|x_{b,j} x_{c,k}) = P(X_a = x_{a,i}|x_{c,k})$, for all events in the sample space of X_a, X_b and X_c. A credal network is *polytree-shaped* if there is only one path between any two nodes in the underlying undirected graph.

An *extension* of a credal network is a joint credal set that can be associated with the network and that satisfies all constraints in the network [Cozman, 2000]. The *strong extension* is the largest credal set that agrees with the strong independence assumptions explicited in the credal net. This extension is given by the convex hull of all distributions that satisfy $\prod_i p(X_i|\rho(X_i))$, where each distribution $p(X_i|\rho(X_i))$ is selected from the extremes points of the local credal set $K(X_i|\rho(X_i)) \in \mathbf{Q}(X_i|\rho(X_i)|)$.

Let $\{X_q = x_{q,i}\}$ be the event of interest in a credal network $\mathbf{C}$. The first algorithms aiming at computation of lower/upper probabilities for $\{X_q = x_{q,i}\}$ employed exhaustive procedures that verified all potential vertices of the strong extension [Tessem, 1992]. For each one of these joint distributions, it is then necessary to compute:

$$P(X_q = x_{q,i}|E) = \frac{\sum_{\mathbf{X}\setminus\{X_q,\mathbf{X}_E\}, X_q = x_{q,i}} \prod_a p(X_a|\rho(X_a))}{\sum_{\mathbf{X}\setminus\mathbf{X}_E} \prod_a p(X_a|\rho(X_a))}. \tag{1}$$

This exhaustive search can be expressed using message propagations schemes [Moral, 1999, Tessem, 1992]; we review this type of scheme for polytree-shaped networks in Section 3. Such schemes generalize similar algorithms for inference in Bayesian networks [Verma and Pearl, 1988]; in fact, one of the advantages of strong extensions is that they entail the same *d-separation* relations that exist in Bayesian networks [Cozman, 1998, Verma and Pearl, 1988]. In spite of its elegance, message propagation schemes for strong extensions are still exhaustive methods that can only handle very small networks. The only exact algorithm that can deal with large networks is the 2U algorithm, that is otherwise restricted to polytree-shaped networks with binary variables [Fagiouli and Zaffalon, 1998].

The limitations of exhaustive methods have motivated research on exact algorithms which do not use enumerative techniques — for example, the branch-and-bound techniques of Rocha and Cozman (geared towards polytree-shaped networks) [Rocha and Cozman, 2003], and the multilinear programming approach of Campos and Cozman [Campos and Cozman, 2004]. This last algorithm can be applied to general credal networks, and it is probably the most efficient exact algorithm currently available. Still, exact algorithms cannot handle large networks; in fact even relatively small networks can offer unsurmountable challenges, depending on the characteristics of the network. The computational cost of inference depends on several factors, such as the number of variables and credal sets in the network, the number of categories of each variable, the number of vertices in each credal set and the network topology.

This situation has led to the proposal of algorithms for approximate inference — that is, algorithms to approximate the upper and lower probabilities. We say that an approximation is an *outer* one if it encloses the probability interval of the event of interest. An approximation is an *inner* one if the approximated interval is enclosed by the exact interval. Several algorithms for approximate inference are available [Cano et al., 1994, Cano and Moral, 1996, Cano and Moral, 2002, Cano and Moral, 1999, Campos and Cozman, 2004, Ide and Cozman, 2004, Rocha et al., 2003, Tessem, 1992].

3. Inference by Decomposition in Subnetworks

In this section we explore decomposition schemes for inference in polytree-shaped credal networks — the idea is to divide a credal network in subnetworks, such that these subnetworks can be processed independently with different trade-offs between time and quality. Such a divide-and-conquer strategy is not new in the context of Bayesian networks, for two reasons. First, inference in large, densely connected Bayesian networks may be very complex. Second, there has been interest in applications of Bayesian networks in embedded systems with very little memory and low processing power [Ramos et al., 2000]. These challenges have led to combinations of exact and approximate algorithms; for example, the use of Gibbs sampling inside clustering algorithms [Kjaerulff, 1994], the combination of clustering and stochastic approximations in dynamic models [Doucet et al., 2000], the use of conditioning inside variable elimination algorithms [Dechter., 1996]. Decomposition schemes have been perfected in the course of that work [Darwiche, 2001]. Recently a synthesis of decomposition schemes and combinations of algorithms has been proposed through the *adaptive conditioning* algorithm [F.T.Ramos and F.G.Cozman, shed].

Even though decomposition/combination schemes have a similar motivation for Bayesian and credal networks, there are significant differences between these networks. For one thing, credal networks are harder to handle exactly; thus it makes sense to have some subnetworks processed exactly, and some processed approximately — whereas in Bayesian networks we can consider exact algorithms for all subnetworks, as long as they are "small" enough. Secondly, polytree-shaped credal networks are already difficult to process, while polytree-shaped Bayesian networks are generally trivial objects from a computational point of view.

In this paper we focus on decomposition schemes that combine exact and approximate methods across subnetworks. The IDS (Inference by Decomposition in Subnetworks) is a direct translation of this proposal: the algorithm divides a polytree in several simpler polytrees; one of these subnetworks contains the query variable, and this subnetwork receives more computational effort than the other subnetworks. Each subnetwork can be viewed as an independent processor that computes an interval-based probabilistic message and sends it to the main subnetwork or to another subnetwork. At the end of this message propagation scheme, the main subnetwork has all information that it needs to compute an approximation to the probability interval of interest.

3.1. Set and interval messages in polytree-shaped credal networks

The IDS algorithm exploits the set-based message propagation algorithm discussed by Moral [Moral, 1999]; we first present a very short review of that algorithm, just to fix notation and terminology. For now we consider full-fledged set-based messages; later we will consider the use of interval messages.

(1) The message propagation algorithm starts when the query node X_q requests that its parents and children send messages to compute $K(X_q|E)$. After receiving answers, the processor associated with X_q calculates the upper and lower probabilities of $\{X_q = x_{q,i}\}$ given E in $K(X_q|E)$.

(2) A parent node X_p calculates the credal set $K(X_p|E_p)$ — taking into account evidence E_p that X_p d-separates from the node X_s that requested a message. This credal set is then associated with a message denoted by $\pi^K_{X_s}(X_p)$, and sent to X_s. To calculate this last message, X_p needs some information stored in the branches of the polytree that it separates from X_s. Node X_p requests to its children and parents nodes, except X_s, the necessary messages. Those messages are obtained recursively by local computation and message propagation.

(3) A children node X_c calculates a message containing a set of likelihood functions $p(E_c|X_s)$. Here E_c denotes the evidence that X_c d-separates from X_s. This message is denoted by $\lambda^K_{X_c}(X_s)$ and computed with a procedure that, initially, requests to every $X_a \in \rho(X_c) \setminus \{X_s\}$ and every $X_b \in \chi(X_c)$ send messages $\pi^K_{X_c}(X_a)$ and $\lambda^K_{X_b}(X_c)$, respectively. After receiving these messages X_c integrates them in $\lambda^K_{X_c}(X_s)$.

Note that this propagation scheme essentially sends set-based messages across arcs of a credal network. Therefore, since we intend to apply a decomposition procedure in credal networks, we must investigate what happens if we "cut" an arc when decomposing a network — in polytree-shaped Bayesian networks, any arc divides a network, and the messages that would flow across the arc carry all the information between the subnetworks [Peot and Shacter, 1991]. The same observation applies to polytree-shaped credal networks with strong independence:

Theorem 1 *Let $\mathbf{C}$ be a polytree credal network that can be divided in two subnetworks $\mathbf{C}_i$ and $\mathbf{C}_j$ by removing the arc connecting a variable $X_i \in \mathbf{C}_i$ to a variable $X_j \in \mathbf{C}_j$ in the original network. Furthermore, let E_i and E_j be the evidence sets related to nodes in $\mathbf{C}_i$ and $\mathbf{C}_j$, respectively, and, X_q be the query variable. If the query variable is in $\mathbf{C}_j$, we can compute the message $\pi^K_{X_j}(X_i)$ in the first subnetwork and send it to the node X_j in $\mathbf{C}_j$, and to obtain tight intervals in $\mathbf{C}_j$. If the query variable is in $\mathbf{C}_i$, we can compute the message $\lambda^K_{X_j}(X_i)$ in the second subnetwork and send it to X_i, and to obtain tight intervals in $\mathbf{C}_i$.*

Proof: Given the propagation message scheme described previously, we have two situations for X_q. If $X_q \in \mathbf{C}_i$, all information related with the evidence E_j that we need to compute upper and lower probabilities on

X_q is stored in the message $\lambda^K_{X_j}(X_i)$; therefore, if we calculate $\lambda^K_{X_j}(X_i)$ previously, we do not need to manipulate $\mathbf{C}_j$ when computing the interest interval in $\mathbf{C}_i$. If $X_q \in \mathbf{C}_j$, all information related with the evidence E_i that we need to calculate upper and lower probabilities on X_q is stored in the message $\pi^K_{X_j}(X_i)$; so, if we calculate $\pi^K_{X_j}(X_i)$ before, we do not need to deal with $\mathbf{C}_i$ when calculating the interest interval in $\mathbf{C}_j$. From this follows that after message propagation, the subnetworks can be processed separately. Additionally, the computation of the message from $\mathbf{C}_j$ to $\mathbf{C}_i$ is independent of E_i and the computation of the message from $\mathbf{C}_i$ to $\mathbf{C}_j$ is independent of E_j. Thus these messages can also be computed independently. QED

The result is illustrated in Figure 2.

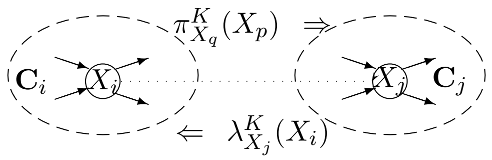


Figure 2: Decomposition of a polytree credal net.

This decomposition makes each subnetwork more manageable, but still the messages can be rather complex objects. To overcome this difficulty, we simply replace the set-based $\pi^K_\cdot(\cdot)$ and $\lambda^K_\cdot(\cdot)$ messages by interval messages. Note that we can then have *tight* interval messages (where the lower and upper probability values are in fact attained by measures) or *non-tight* interval messages that may be produced by approximate inference algorithms in the subnetworks.

3.2. Implementing IDS

In our implementation of the IDS algorithm, a network is divided by a depth-first search based procedure on the graph of $\mathbf{C}$. The basic parameter of this procedure is a integer L that specifies the maximum size of the subnetwork that can be processed exactly (we take the MLR algorithm to be our standard "exact" method [Campos and Cozman, 2004]).

The depth-first procedure starts in the query node and crosses the network, marking visited nodes. Every time a node X_i is visited the number of combinations of vertices in the current subnetwork, W, is updated. If W exceeds L the node X_i is disconnected from the node X_j (the direct ascendent of X_i in the graph). Figure 3.2 contains a sketch of the decomposition procedure (the network and the threshold L are global variables). The function $W(X)$ calculates the number of combinations of vertices in the subnetwork formed by the nodes already visited. After decomposition, we say that an ancillary subnetwork $\mathbf{C}_a$ is *below* the main subnetwork if there is an arc in the original credal network that connects one node in the main subnetwork to one node in $\mathbf{C}_a$. Inversely, we said that $\mathbf{C}_a$ is *above* the main subnet if there is an arc in the original credal network that connects one node in $\mathbf{C}_a$ to one node in the main subnetwork.

Since the original credal network is divided the IDS algorithm computes messages in the ancillary subnetworks and to send them to the main subnetwork. The interval-based message that the main subnetwork $\mathbf{C}_q$ receives from a subnetwork $\mathbf{C}_a$, above it, is denoted by $\pi^d_{\mathbf{C}_q}(\mathbf{C}_a)$. The interval-based message that the main subnetwork receives from an ancillary subnetwork $\mathbf{C}_b$, below it, is indicated by $\lambda^d_{\mathbf{C}_b}(\mathbf{C}_q)$. Currently, these messages are calculated with the approximate A/R++ algorithm [Campos and Cozman, 2004]. Let $X_a \in \mathbf{C}_a$ be the variable that was disconnected to a node $X_c \in \mathbf{C}_q$. The use of the A/R++ algorithm to calculate $\pi^d_{\mathbf{C}_q}(\mathbf{C}_a)$ results that set of intervals composing this message are

Input: The current node X_i and the last node visited X_j.
Output: Set of subnetworks obtained by removing arcs.

- If $W(X) > L$, then if there is an arc connecting X and Y in $\mathbf{C}$, remove it;
- else,
  - select a non-visited node $X_k \in \rho(X_i)$ and run the decomposition procedure on it;
  - select a non-visited node $X_l \in \chi(X_i)$ and run the decomposition procedure on it.

Figure 3: Decomposition in the IDS algorithm.

outer bounds for the lower and upper probabilities of every category of X_a given the evidence E_a. Similarly, let $X_b \in \mathbf{C}_b$ be the variable that was disconnect from a node $X_p \in \mathbf{C}_q$. The intervals of the message $\lambda^d_{\mathbf{C}_b}(\mathbf{C}_q)$ are outer bounds for the likelihood functions as $p(E_b|X_p = x_{p,j})$, for all $\{X_p = x_{p,j}\}$ in the sample space of X_p.

After to receive the requested messages the subnetwork $\mathbf{C}_q$ converts their intervals in sets of probability functions. The intervals in $\pi^d_{\mathbf{C}_q}(\mathbf{C}_a)$ are used to generate the largest credal set that agrees with the intervals of that message, this credal set is denoted as $K'(X_a|E_a)$. In sequence, a new node labelled X'_a is added to $\mathbf{C}_q$ and it is made a parent of X_c. This new node is equivalent to X_a and its collection of separated credal sets contains an unique element, the credal set $K(X'_a)$ equivalent to $K'(X_a|E_a)$. Similarly, every message $\lambda^d_{\mathbf{C}_b}(\mathbf{C}_q)$ that X_q receives is converted in the largest convex set of likelihood functions as $p(E_c|X_p)$ that agrees with the intervals in the message. In next, this set is associated to X_p as a dummy evidence [Pearl, 1988] [Tessem, 1992].

This procedure transforms the subnetwork $\mathbf{C}_q$ in a new polytree $\mathbf{C}'_q$ in which it is possible to compute intervals for the event of interest. Given it, the IDS algorithm applies an exact method, MLR, to calculate the upper and lower probabilities of $\{X_q = x_{q,i}\}$ in $\mathbf{C}'_q$. However, some credal sets in $\mathbf{C}'_q$ are the larger than those credal sets that are dealt by the exact message propagation algorithm. From this follow that the problem of the inference of the probability interval of $\{X_q = x_{q,i}\}$ in $\mathbf{C}'_q$ is a relaxed version of the problem of the inference in $\mathbf{C}$ and the extreme probabilities obtained by IDS are outer bounds.

The combination of the MLR and A/R++ algorithms proposed by the IDS algorithm is interesting for two main reasons. First, because the MLR algorithm already runs the A/R++ in a preprocessing phase. Therefore, the initial execution of the A/R++ algorithm on the ancillary subnetworks has no impact on the overall computational cost. Second, because the MLR algorithm is an anytime procedure. An algorithm is anytime if it can produce a solution in a given time T and the quality of solutions improve with time after T [F.Ramos et al., 2002]. It means that when computing intervals for $\{X_q = x_{q,i}\}$ in $\mathbf{C}'_q$ we can use the MLR to calculate exact or approximate ones. The selection of one of these strategies depend on the time and resources avaliable.

4. Example

To illustrate the characteristics of the IDS algorithm, consider the following (rather large) problem. We have a dynamic network [Russell and Norvig, 1995] formed by replication of a network slice depicted in Figure 4, where all variables have three values. The objective here is to compute lower and upper probabilities for the values of variable X_9 in the last slice; we considered twelve time slices — the inference requires the manipulation of

84 variables, and the number of potential vertices of the strong extension is 10^{355}.

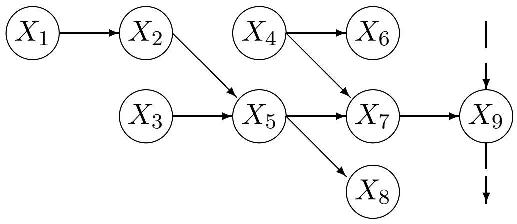


Figure 4: A time slice of the credal nets used in tests.

In an inital experiment showed that this inference cannot be computed exactly with the MLR algorithm. So, we approximate the intended intervals with the IDS algorithm. To calculate this approximation we run IDS as described above - the main subnetwork was exactly processed by MLR algorithm. The parameter L of the IDS algorithm was set as 5.5×10^{15}, what makes that the main subnework was divided into two not balanced subnetworks. The first subnetwork consisted of the last time slice, while the second subnetwork consisted of all other time slices. The relative error in probability intervals computed by the IDS algorithm were smaller than 1.6%, while the largest error of the fast combined framework that executes A/R++ and a iteration of MLR algorithm was 1.7%. The smallest error of IDS was 0.8% while the smallest error of the combined framework was 0.47%. The time spent by IDS to compute the inference was between 0.7 and less than 3 times the time spent by the combined framework.

The experiment indicates that IDS provides a compenting strategy to approach approximate inferences that can not be solved with MLR. Furtherly, the IDS algorithm also can be viewed as a method for anytime inference. In the IDS this behaviour can be reached by increasing the threshold L. That is, if we increase the threshold we are, probably, allowing that more time be spent in the computation of the main subnetwork, therefore, we are allowing that a more complex subnetwork be dealt exactly, and, it tends to improve the precision of the algorithm [da Rocha, 1997]. Alternatively, by reducing L we obtain a faster inference procedure at the expense of precision.

5. Conclusion

This paper presented an approximate algorithm called IDS that computes outer approximations for probability intervals in credal networks. The main characteristic of this method is that it provides a simple strategy to manage the tradeoff between the precision of the calculated intervals and the cost for to compute an inference. For that, the algorithm uses a divide-and-conquer approach and a message propagation scheme that allows to combine an approximate exacts inference algorithms.

Furthermore, the decomposed inferences are solved with different algorithms. Partial results are computed with approximate methods while the main result is obtatined with a combination of the partial results and exact or approximate inference algorithms. The experiment indicates that such procedure allows to trade off time and precision when computing probability intervals in an anytime scheme.

Future projects considers the extension of this framework to inferences in multiply connected credal networks.

References

Campos, C. P. and Cozman, F. G. (2004). Inference in credal networs using multilinear programming. In *Second Starting AI Researcher Symposium (STAIRS)*, pages 50–61, Valencia, Spain. IOS Press.

Cano, A., Cano, J. E., and Moral, S. (1994). Convex sets of probabilities propagation by simulated annealing. In Bouchon-Meunier, B., Yager, R., and Zadeh, L., editors, *5th Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems*, volume 945 of *Lecture Notes in Computer Science*, pages 978–983, Paris, França. Springer.

Cano, A. and Moral, S. (1996). A genetic algorithm to approximate convex sets of probabilities. In Dambrosio, B. and Moral, S., editors, *7th Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems*, volume 2, pages 859–864, Granada, Espanha. Universidade de Granada, DECSAI, Universidade de Granada.

Cano, A. and Moral, S. (1999). A review of propagation algorithms for imprecise probabilities. In Cooman, G., Cozman, F., Moral, S., and Walley, P., editors, *1st International Symposium on Imprecise Probabilities and Their Applications*, pages 51–60, Ghent. Society for Imprecise Probability Theory and Applications.

Cano, A. and Moral, S. (2002). Using probability trees to compute marginals with imprecise probabilities. *International Journal of Approximate Reasoning*, 29(1):1–46.

Couso, I., Moral, S., and Walley, P. (1999). Examples of independence for imprecise probabilities. In Cooman, G., Cozman, F., Moral, S., and Walley, P., editors, *1st International Symposium on Imprecise Probabilities and Their Applications*, pages 121–130, Ghent. Society for Imprecise Probability Theory and Applications.

Cozman, F. G. (1997). Robustness analysis of Bayesian networks with local convex sets of distributions. In Geiger, D., Shenoy, P., and Horvitz, H., editors, *13th Annual Conference on Uncertainty in Artificial Intelligence Conference*, San Francisco. Morgan Kaufmann.

Cozman, F. G. (1998). Irrelevance and independence relations in Quasi-Bayesian networks. In Cooper, G. and Moral, S., editors, *14th Annual Conference on Uncertainty in Artificial Intelligence Conference*, pages 86–96, Madison, Wisconsin. Morgan Kaufmann.

Cozman, F. G. (2000). Credal networks. *Artificial Intelligence*, 120(2):199–233.

da Rocha, J. C. F. (1997). *Algoritmos para Inferência em Redes Credais*. Doutorado, Universidade de São Paulo, São Paulo, Brasil. em português.

Darwiche, A. (2001). Recursive conditioning. *IArtificial Intelligence*, 126(1-2):5–41.

Dechter., R. (1996). Topological parameters for time-space tradeoff. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 220–227.

Doucet, A., de Freitas, N., Murphy, K., and Russell, S. (2000). Rao-blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 176–183.

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

Fertig, K. W. and Breese, J. S. (1990). Interval influence diagrams. In Henrion, M., Shacter, R., Kanal, L., and Lemmer, J., editors, *6th Annual in Artificial Intelligence Conference*, volume 5, pages 149–161, Amsterdam. North-Holland.

F.Ramos, Cozman, F. G., and Ide, J. S. (2002). Embedded bayesian networks: Anyspace, anytime probabilistic inference. In *AAAI/KDD/UAI Workshop in Real-time Decision Support and Diagnosis Systems*, pages 118–127.

F.T.Ramos and F.G.Cozman (To published). Anytime, anyspace probabilistic inference. *International Journal of Approximate ReasoningArtificial Intelligence*.

Ha, V. A., Doan, A., Vu, V., and Haddawy, P. (1998). Geometric foundations for interval-based probabilities. *Annals of Mathematics and Artificial Intelligence*, 24(1):582–593.

Ide, J. S. and Cozman, F. G. (2004). Ipe and l2u: Approximate algorithms for credal network. In *Second Starting AI Researcher Symposium (STAIRS)*, pages 118–127. IOS Press.

Kjaerulff, U. (1994). Combining exact inference and Gibbs sampling in junction trees. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 368–375.

Levi, I. (1980). *The Enterprise of Knowledge*. MIT Press, Cambridge.

Moral, S. (1999). Algorithms for imprecise probabilities. Technical report, Universidade de Granada, DCCIA, Universidade de Granada.

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

Peot, M. A. and Shacter, R. D. (1991). Fusion and propagation with multiple observations in belief networks. *Artificial Intelligence*, 48(3):299–318.

Ramos, F., Mikami, F., and Cozman, F. G. (2000). Implementação de redes Bayesianas em sistemas embarcados. In *Proceedings of the IBERAMIA/SBIA 2000 Workshops (Workshop on Probabilistic Reasoning in Artificial Intelligence) - in Portuguese*, pages 65–69. Editora Tec Art.

Rocha, J. C. F., Campos, C. P., and Cozman, F. G. (2003). Inference in polytrees with sets of probabilities. In *19th Annual Conference on Uncertainty in Artificial Intelligence Conference*, pages 217–224, San Francisco. Morgan Kaufmann.

Rocha, J. C. F. and Cozman, F. G. (2003). Inference in credal networks with branch-and-bound algorithms. In *3rd International Symposium on Imprecise Probabilities and Their Applications*, pages 482–495, Lugano, Suíça. Society for Imprecise Probability Theory and Applications.

Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A modern approach*. Prentice Hall, Upper Saddle River.

Tessem, B. (1992). Interval probability propagation. *International Journal of Approximate Reasoning*, (7):95–120.

Verma, T. and Pearl, J. (1988). Causal networks: semantics and expressiveness. In Levitt, T., Schacter, R., Lemmer, J., and Kanal, L., editors, *4th Annual Conference on Uncertainty in Artificial Intelligence*, pages 352–359, Mineapolis. North Holland.

Walley, P. (1991). *Statistical Reasoning with Imprecise Probabilities*. Monographs on Statistics and Applied Probability. Chapman and Hall, London.

Wellman, M. P. (1990). Fundamental concepts in qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303.

Zaffalon, M. (1998). Credal classifier. *Artificial Intelligence*, 106(1):77–107.