Algorithms for Approximated Inference with Credal Networks

Jose Carlos F. da Rocha[1,2], Cassio P. de Campos[1,3], and Fabio G. Cozman[1]

[1] Decision Making Lab
Escola Politécnica, University of São Paulo
Av. Prof. Mello Moraes, 2231
05508-900, São Paulo, SP, Brazil
http://www.pmr.poli.usp.br/ltd/
[2] Departamento de Informática
Ponta Grossa State University
Av.Carlos Cavalcanti, 4647
84013-900 Ponta Grossa, PR, Brazil
jrocha@uepg.br
[3] Departamento de Ciência da Computação
Pontifical Catholic University of São Paulo
São Paulo, SP, Brazil

Abstract. A credal network associates convex sets of probability distributions with graph-based models. Inference with credal networks aims at determining intervals on probability measures. Here we describe how a branch-and-bound based approach can be applied to accomplish approximated inference in polytrees iteratively. Our strategy explores a breadth-first version of branch-and-bound to compute outer approximations for the probability intervals. The basic idea is to refine the outer bounds calculated by the A/R+ algorithm until they are sufficiently precise or time/memory constraints have been exceeded ...

1 Introduction

The Credal Network formalism represents imprecise and uncertain probabilistic knowledge through direct acyclic graphs (DAG) [11]. Each node in a credal net represents a random variable and the topology of the DAG highlights independence assumptions; each variable is independent of its nondescendants nonparents given its parents. As a Bayesian network [18], the variables of a credal net are associated with probabilistic data that encodes the strength of the relationships. In a credal network, this knowledge is encoded by attaching closed convex sets of probability distributions to each node. These sets are called credal sets. Basically, a credal set quantifies the uncertainty and imprecision on probabilistic measures [22].

A credal network can thus be viewed as a Bayes net with relaxed numerical statements. In this context, credal networks can be an useful tool to: (a) execute sensitivity analysis of Bayes nets [9]; (b)represent vague probabilistic

statements[23]); (c) deal with conflicting preferences [10]. While the independence assumptions in a Bayes net lead us to represent a joint distribution using conditional probabilities, the independence assumptions of a credal net lead to a credal set of joint distributions with credal sets of conditional distributions. However, there are several independence concepts that can be defined on a collection of credal sets and different concepts lead to different joint credal sets. Basically, the selection of one concept depends on the application. In this paper we interpret the independence relations in a credal net as statements of *strong independence* [13] [11]. Our interest in this kind of independence is due to two facts. First, its semantic is reasonable for robustness analysis of Bayes nets[9]. Second, it allows us to apply the *d-separation* criteria [18] to detect what variables are relevant to the inference.

Algorithms for inference with credal nets compute an interval for the probability of each category of a *query* variable. The extremes of this interval are called the *upper* probability and the *lower* probability and their computation requires optimization. The computation of a upper (or lower) probability can be viewed as a signomial program under linear constraints [21]. From an algorithmic point of view, this problem is NP-hard even for simple models as polytrees (DAG without loops). Exact and approximate algorithms have been proposed in the literature. Exact algorithms calculate the correct value for upper and lower probabilities while approximated algorithms calculate an outer or inner estimative. An outer approximation contains the exact interval while an inner approximation is enclosed by the exact interval.

In this paper we propose a new approach for approximate inference in polytrees, the use of a breadth-first branch-and-bound procedure to improve a given outer approximation. The basic idea is to compute iteratively tighter bounds from an initial value. This approach extends our previous work, which is described in [15] and [17], where we advocated the use of a depth-first branch-and-bound algorithm to compute exact or inner bounds in polytrees. In those papers the tests showed that depth-first branch-and-bound can be effective but it can not to deal with arbitrarily complex inferences of exact values. It motivated us to extend that approach to compute outer bounds.

The start point of the breadth-first branch-and-bound algorithm is the outer approximation computed by the A/R+ algorithm[15], an algorithm for polytrees. The branch-and-algorithm also explores the A/R+ as an heuristic to guide the search. Furthermore, an algorithm that calculates inner approximations through multilinear programming techniques, described in [15], is used as a pruning heuristic.

The organization of the text is as follows. Section 2 presents a summary of credal networks. Section 3 illustrate the main ideas about the A/R+ algorithm and about the inference algorithm that is based on multilinear programming techniques. Section 4 describes how a breadth-first branch-and-bound can be used to compute outer approximations for inferences with credal nets. Finally, we discuss the results and highlight our main ideas about future work.

2 Credal sets, credal networks and inference

Given a random variable X, a credal set $K(X)$ is a closed convex set of probability densities on X [14][4]. In this paper we assume that a credal set has a finite number of vertices and we can represent such a set just enumerating its extreme distributions: $K(X) \equiv ch[p_0(X), \ldots, p_s(X)]$ ($ch[\cdot]$ denotes the convex hull operation [12]). A conditional credal set is a set of conditional distributions; for example, $K(Y|X = x) \equiv ch[p_0(Y|X = x), \ldots, p_t(Y|X = x)]$. We can obtain the conditional credal set applying Bayes rule to each distribution in a credal set of joint distributions. Let X to have m categories, given $K(X)$ and $K(Y|X) = \{K(Y|x_0), ..., K(Y|x_m)\}$, an *extension* $K(XY)$ from those sets is a credal set of joint distributions with given marginal and conditional credal sets.

When the credal set $K(X)$ is known, we can relate each category x of X with a probability interval $[\underline{p}(x), \overline{p}(x)]$. Here $\underline{p}(x) = \min(p(x) : p(X) \in K(X))$, is called lower probability and $\overline{p}(x) = \max(p(x) : p(X) \in K(X))$, is the upper probability. Walley interprets theses values as the minimum/maximum betting rate on/against x [26].

Credal networks associate credal sets with a DAG that indicates a number of independence relations. Every node X_i in the DAG is associated with a variable[5], and every variable is associated with a list of *local* credal sets $K(X_i|\mathrm{pa}(X_i))$, where $\mathrm{pa}(X_i)$ denotes the parents of X_i in the graph. So, a node stores a list of credal sets $\{K(X_i|pa(X_i)_0), ..., K(X_i|pa(X_i)_p)\}$, where $pa(X_i)_0, ..., pa(X_i)_p$ are the instances of $pa(X_i)$. If X_i is a root node, its credal set is $K(X_i)$. When the credal sets of $K(X_i|pa(X_i))$ have no relationship with each other for different values of $pa(X_i)$ they are called *separately specified*. In this paper we assume that local credal sets are always separately specified.

The basic assumption in a credal net is that every variable is independent of its nondescendants nonparents given its parents. The semantics of such a condition depends on which concept of independence is adopted [11] [13] [20]. In this paper we adopt the concept of strong independence: two variables X and Y are strongly independent when every extreme point of $K(X, Y)$ satisfies stochastic independence of X and Y [25]. That is, each vertex $p(X, Y) \in K(X, Y)$ factorizes as $p(X, Y) = p(X) \cdot p(Y)$, given $p(X) \in K(X)$ and $p(Y) \in K(Y)$. The *strong extension* of a credal network is the largest joint credal set such that every variable is strongly independent of its nondescendants nonparents given its parents. The strong extension of a credal network is the joint credal set that is equivalent to the convex hull of every possible combination of vertices for all credal sets in the net [11]. Each vertex of a strong extension factorizes as follows:

$$p(X_1, \ldots, X_n) = \prod_i p(X_i|\mathrm{pa}(X_i)) \,. \tag{1}$$

Figure 1 shows an example of a credal network structure.

[4] We deal only with convex sets.

[5] To simplify the text, we represent a node and its variable with the same symbol.

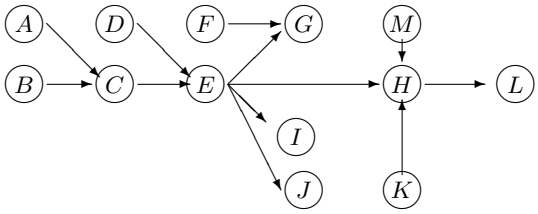

Fig. 1. A polytree credal network.

An *inference* in a credal network calculates tight bounds for probability values (the upper and lower probabilities) in an extension of the network. If X_q is a *query* variable with categories given by $(x_{q_0}, x_{q_1}, ...x_{q_f})$, and $\mathbf{X}_E$ represents a set of *observed* variables, then an inference determines the interval $[\underline{p}(x_{q_k}|\mathbf{X}_E), \overline{p}(x_{q_k}|\mathbf{X}_E)]$ for each $k \in \{0, .., f\}$. The difficulty faced by inference algorithms is the potentially enormous number of distributions that a strong extension can have. Consider the next example [16].

Example 1 Consider a network with four variables X, Y, Z and W; W is the sole child of X, Y and Z. Suppose that all variables are ternary and that every local credal set has three vertices. If we want to compute the strong extension $K(X, Y, Z, W)$, we must consider the fact that its vertices are given by $p(W, X, Y, Z) = p(W|X, Y, Z) \cdot p(X) \cdot p(Y) \cdot p(Z)$. Now, W is associated to 27 credal sets, and therefore there are 3^{27} ways to combine the vertices of these credal sets to compose $p(W|X, Y, Z)$. Besides, these 3^{27} points must be combined with the 3^3 combinations of the vertices from the *priors*. So, the potential number of vertices in $K(X, Y, Z, W)$ is 3^{30}.

Most exact algorithms are based on enumerative schemes. Given the previous example we can see that this is not a promising approach, even more if we have no efficient pruning scheme. On the other hand, approximated algorithms explore the application of iterative improvements, local optimization or interval valued representations to avoid enumeration. Similar strategies are explored by next algorithms.

3 Algorithms for inner and outer approximations

In this section we discuss two approximated algorithms. The first algorithm, A/R+, computes outer bounds and it will be used as an heuristic function by the branch-and-bound procedure described in the section 4. The second algorithm applies multilinear programming to compute inner bounds and will be used as a heuristic for prune in the section 4. These algorithms are discussed in [15].

3.1 Computing outer approximations with the A/R+ algorithm

The A/R+ algorithm is an extension of the Tessem's A/R algorithm [5] that computes outer intervals for $p(X_q|\mathbf{X}_E)$ in an *interval valued* Bayes net. An interval valued Bayes net is a Bayesian network that contains intervals of probabilities associated with its numerical parameters. So, given a node X_i, each entry $p(X_i = x|pa(X_i)_k)$ of its conditional probability table is an interval denoted as $[p(X_i = x|pa(X_i)_k)_*, p(X_i = x|pa(X_i)_k)^*]$.

The A/R algorithm can be applied to compute inferences with credal nets. To make that we convert a credal network to an interval valued Bayes net by producing probability intervals from each $K(X_i|pa(X_i)_k)$. The extremes of these intervals are calculated as lower and upper bounds. Credal sets and intervals of probability do not represent the same thing. Basically, intervals of probability provide a relaxed and wider representation for credal sets.

The A/R algorithm extends Pearl's message propagation scheme (an inference algorithm for Bayes nets) for interval valued polytrees. The propagation scheme is basically the same but all messages and vectors dealt by A/R are interval valued, that is, *bel*, λ, π, $\pi_Y(X)$, $\lambda_X(y)$ are vectors of probability intervals. The procedures that compute the intervals of these vectors and messages are based on an interval arithmetic and an optimization technique, named *annihilation* and *reinforcement* (thus the name A/R). This technique is recommended when the objective function implies a sum of products taking in consideration normalized quantities [7], and this is the case here.

To give an idea of the mechanics of the A/R algorithm we describe below one of its procedures. We show how to compute the lower bound $\pi_*(x_i)$ for the interval valued function $\pi(X)$. That procedure computes $\pi_*(x_i)$ for a node X with parents $Y_0, ..., Y_k$. Similar operations compute the upper bound $\pi^*(x_j)$.

1. Construct the interval-valued function $\beta(Y_0, \ldots, Y_k)$ by interval multiplication of the messages $\pi_X(Y_i)$ received by X (these messages are also interval-valued).
2. Construct a distribution $p(Y_0, \ldots, Y_k)$, consistent with $\beta(Y_0, \ldots, Y_k)$, such that $p(Y_0, \ldots, Y_k)$ minimizes $\pi_*(x_j) = \sum_{Y_0,\ldots,Y_k} \underline{p}(x_j|Y_0, \ldots, Y_k)\, p(Y_0, \ldots, Y_k)$, where $\underline{p}(x_j|Y_0, \ldots, Y_k)$ is the lower value for $p(x_j|Y_0, \ldots, Y_k)$.

These operations are efficient. The first step is a product of extreme values of the local interval messages (it concerns just to a node and its parents). The computation of $p(Y_0, \ldots, Y_k)$ is not hard either; sort $\underline{p}(x_j|Y_0, \ldots, Y_k)$ in increasing order and distribute the probability mass (consistently with $\beta(Y_0, \ldots, Y_k)$) from the smallest to the largest value of $\underline{p}(x_j|Y_0, \ldots, Y_k)$. Similar procedures are applied to compute $\lambda_X(Y_i)$ and $\pi_Z(X)$ messages (where Z is a child of X), while the function $\lambda(X)$ is obtained by interval multiplication.

The outer approximations calculated by A/R can be wide or even unacceptable (very wide) [3]. The A/R+ algorithm improves these approximations using a similar message propagation scheme. In the A/R algorithm the proposal is to

replace some calculations based on interval arithmetic by exhaustive computations if time and memory constraints allow it. To make that, the procedures that compute $\pi(X)$ and $\lambda_X(Y_i)$ were modified.

We can have an idea of the differences between A/R and A/R+ just studing how A/R+ extends the previously commented algorithm. A fundamental change occurs in the first step of the algorithm that is substituted by an enumerative scheme that divides the first step in two new steps: (a) for each $\pi_X(Y_i)$ calculate the largest credal set that matches with it; (b) replace $\beta(Y_0, \ldots, Y_k)$ with the strong extension from the generated credal sets. In sequence, this extension is used in the optimization procedure of the second step.

Previous research showed that A/R+ is more precise than A/R [15].

3.2 Inner approximations with multilinear programming

An inner approximation for $\overline{p}(X_q|\mathbf{X}_E)$ can be generated by any method that looks for a local maxima of $p(X_q|\mathbf{X}_E)$ subject to constraints imposed by local credal sets $K(X_i|\text{pa}(X_i))$. Such approximations have been considered in previous literature, using for example simulated annealing [1] and genetic algorithms [2]. Generally these methods require tuning several parameters. Other ideas, such as gradient search or geometric programming, have also been proposed but not implemented so far [9] [24] — such techniques would demand a great deal of numerical finesse and would have to be adapted carefully to credal networks.

The computation of upper or lower bounds is a maximization problem whose objective function is a ratio of two large multilinear functions represented by (1) (such a problem is typically classified as a *signomial* program [21]). In [15] we present a local search algorithm that exploits the fact that, in our problem, a) all constraints are linear; b) all constraints are in some sense "local" (they are grouped with the local credal sets $K(X_i|\text{pa}(X_i))$); and c) multilinear functions are possibly the simplest signomial functions to be found. The algorithm does benefit from these properties, with no free parameters to tune, and not requiring special methods to control numerical errors.

The algorithm described in [15] is based on the Lukatskii-Shapot's algorithm for local search in multilinear programs [4]. The tests in that paper have indicated that it can produce *extremely* accurate approximations.

4 Iterative outer approximation: The breadth-first branch-and-bound

The previous sections suggest the following scenario. Suppose that someone needs to make decisions based on $\overline{p}(X = x)$, but to compute this bound is very expensive. An alternative would be to compute the interval of the approximations for $\overline{p}(X = x)$, $[\hat{p}(X = x), p^+(X = x)]$, and to use this interval to support the decision making. A similar process can be useful when making decisions based on lower probabilities.

It is likely that many applications will find the precision of these approximations satisfactory. But, what happens if the user needs more precise answers? Would be possible to improve an estimative using a procedure that considers time and memory restrictions? In this section we approach these questions with a breadth-first branch-and-bound procedure. The proposal is to use branch-and-bound to improve the A/R+ bounds in an iterative way; the algorithm iterates until the approximation be sufficiently accurate or to exceed time or memory constrains. The first factor can be evaluated by verifing the amplitude of the interval between outer and inner approximations. The second factor can be implemented by defining a limit to the maximum number of iterations of the branch-and-bound algorithm.

Given a maximization problem P with an objective function, a breadth-first branch-and-bound procedure divides the feasible region of P in sub-regions [8]. These sub-regions must produce sub-instances of P. These sub-instances must be easier to solve or approximate than P is and the solution for P must be present in one of them. Each sub-instance R is evaluated with a relaxed algorithm that produces an overestimated bound $r(R)$ and an underestimated bound $s(R)$. This procedure is repeated while there is a new promising alternative for partitioning. The optimum is the best singleton instance found before the algorithm to stop. We can view the brand-and-bound procedure as a search algorithm. In this case, the r bound can be considered as a search heuristic and the s bound as a prune heuristic. In the next we show our implementation of breadth-first branch-and-bound algorithm [6].

1. Input : the whole problem P with a objective function $f(x)$ defined on $\mathbf{X}$.
2. Ancillary data structures: a list named OPEN, initially OPEN contains just P; the best value found until the moment, named UPPER;
3. (Any problem instance must be evaluated by two relaxed functions r and s; these functions produces outer and inner estimations, respectively.)
4. Repeat:
   (a) remove an instance Y from OPEN;
   (b) divide Y in sub-instances Y_0,Y_1,...;
   (c) for each Y_j
      i. if s(Y_j) $\leq$ UPPER, then place Y_j in OPEN; if in addition r(Y_j) $<$ UPPER then set UPPER = r(Y_j), and if Y_j is a singleton, mark it as the best solution found so far;
   (d) if OPEN is not empty, go to step (a), else terminate;

In our application the whole problem is to produce the inference with the entire network; simplified problems are generated by dividing the original network in sub-instances. A sub-instance of a credal network is generated by fixing one vertex of one local credal set at each step. This procedure can be viewed as a search tree where a root node represents the whole problem (with the complete credal net), and where the leaves are standard Bayesian Networks. Intermediary nodes represent sub-instances of the original problem where some vertices were fixed. Furthermore, when computing approximations for upper probabilities, we applied the A/R+ algorithm as an overestimation for $p(X_q = x_q)$, so

$p^+(X_q = x_q)$ implements $r(R)$. Respectively, we use $\hat{p}(X_q = x_q)$ as the underestimated bound $s(R)$. When a leaf node is reached, we run a standard inference algorithm for Bayes nets to obtain an exact probability value. If this value is better than the best value found until the moment, the last is updated.

In [17] we described how branch-and-bound can be applied to produce exact inferences. However, it is possible that a branch-and-bound algorithm needs to evaluate many points of the search space before finding the optimum. That is, sometimes the branch-and-bound algorithm cannot prune the search space significantly. Furthermore, if a great number of nodes need to be expanded, it may be difficult to manage the list OPEN. In [17] we used a depth-first branch-and-bound to deal with this difficult, but that algorithm also can have problems with the size of the search space. Finally, maybe the user has no sufficient time or patient to wait until the inference procedure finishes.

As we said, we approach this question by stoping the branch-and-bound process after to evaluate a predetermined number of points of the search space. When it happens we search in the OPEN list for the element with the maximum A/R+ bound. If this new bound is tighter than the original A/R+ approximation, we set this new value as the inference result. The resultant bound is indicated by $p^{\uparrow}(X_q = x_q)$. The same process can be applied to compute the lower approximation, which is indicated by $p_{\downarrow}(X_q = x_q)$.

We have made experiments with this algorithm. Initially, we took the network of the Figure 1 and produced inferences for $\{E = e_0\}$ and $\{G = g_0\}$ and $\{H = h_0\}$ on randomly generated networks [19]. We generated a number of samples for networks with ternary variables and three vertices by credal set, ternary variables and four vertices by credal set and quaternary variables and three vertices by credal set. Then we evaluated the mean of the difference between p^+ and $\hat{p}$, and the mean of the difference between $p^{\uparrow}$ and $\hat{p}$. The branch-and-bound procedure was interrupted whenever the number of evaluated nodes exceeded 25,000 nodes (for some case the algorithm obtained the exact result before that).

The results of these experiments are showed in the Table 1. The first four columns of the table summarize the experimental conditions. The fifth column shows the mean of the difference between the outer approximations computed by A/R+ and the inner approximations computed by the multilinear programming based algorithm. The last column shows the mean of the difference between the outer approximations computed by branch-and-bound and the inner approximations computed by multilinear programming. There we can see that the use of the branch-and-bound algorithm can effectively reduce the amplitude of the interval between inner and outer bounds.

We have made an additional experiment. We generated five network topologies with fifteen nodes and ternary variables. For every topology we randomly generated three different networks. Each credal set of these networks had three vertices. After that, we run the same inference algorithms - the branch and bound procedure stopped after to explore 10000 nodes of the search space. We proceed similarly to the first experiment and evaluated the intervals for every leaf node.

Table 1. The mean of the amplitude between outer and inner approximations for $\overline{p}(E = e_0)$, $\overline{p}(G = g_0)$ and $\overline{p}(H = h_0)$. The branch-and-bound procedure stopped after evaluating 25,000 nodes of the search space.

♯ categories per variable	♯ vertices per credal set	Potential size of the strong extension	♯ of sample networks	Amplitude $(p^+ - \hat{p})$ (mean)	Amplitude $(p^\uparrow - \hat{p})$ (mean)
Means for $E = e_0$					
03	03	3^{21}	16	0.006	0.00008
03	04	2^{42}	10	0.024	0.011
04	03	3^{35}	10	0.035	0.025
Means for $G = g_0$					
03	03	3^{32}	16	0.016	0.011
03	04	2^{64}	10	0.021	0.012
04	03	3^{35}	10	0.022	0.017
Means for $H = h_0$					
03	03	3^{50}	16	0.017	0.007
03	04	2^{100}	10	0.014	0.009
04	03	3^{101}	04	0.035	0.029

Once again we have observed that the bounds were significantly improved - the mean of the amplitude of the interval $[\hat{p},p^+]$ was 0.012 against 0.003 for $[\hat{p},p^\uparrow]$.

5 Conclusion

We have presented a new approach for approximated inference in credal networks. The approach explores a breadth-first branch-and-bound algorithm to calculate improved outer bounds. The algorithm starts with an A/R+ bound and works iteratively to compute a more precise bound. Furthermore, the search and pruning heuristics of the branch-and-bound procedure are implemented on two approximated inference algorithms. The search heuristic is based on the A/R+ and the prune heuristic is built on an inference algorithm that is based on multilinear programming techniques. The experiments show this approach can be very useful.

Usually, it is necessary to limit the computational effort applied to solve a problem. This was implemented by limiting the number of nodes of the search space that are expanded by branch-and-bound. The definition of this limit depends on factors as the size of the network, the desired precision, time constraints and memory restrictions.

We focus our future work on developing strategies that combine these algorithms in an integrated framework. The basic idea is to divide the problem in parts, to solve each part with a specific algorithm, and to combine the partial results to obtain a complete solution.

References

1. A.Cano, J.E.Cano, and S.Moral. Convex sets of probabilities propagation simulated annealing. In *5th Int. Conf. IPMU-94*, pages 4–8, 1994.
2. A.Cano and S.Moral. A genetic algorithm to approximate convex sets of probabilities. *7th Int. Conf. IPMU-96*, pages 4–8, (Paris, France, July, 1994) 1996.
3. A.Cano and S.Moral. Using probabilities trees to compute marginals with imprecise probabilities. Tech. Rep. DECSAI-00-02-14, University of Granada, Spain, 2000.
4. A.M.Lukatskii and D.V.Shapot. Problems in multilinear programming. *Computational Mathmatics and Mathmatical Physics*, 41(5):638–648, 2001.
5. B.Tessem. Interval probability propagation. *IJAR*, (7):95–120, 1992.
6. C.Papadimitriou and I.Steiglitz. *Combinatorial Optimization, Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
7. D.L.Draper and S.Hanks. Localized partial evaluation of belief networks. In *11th UAI Conference*, pages 170–177. Morgan Kaufmann, 1995.
8. D.P.Bertsekas. *Dynamic Programming, Deterministic and Stochastic Models*. Prentice Hall, Englewoods, 1987.
9. F.G.Cozman. Robustness analysis of bayesian networks with local convex sets of distributions. In *13th UAI Conference*, pages 108–115. Morgan Kaufmann, 1997.
10. F.G.Cozman. A brief introduction to the theory of sets of probability measures. Tech. rep., CMU, December 1999.
11. F.G.Cozman. Credal networks. *Artificial Intelligence*, 120(2):199–233, 2000.
12. H.Edelsbrunner. *Algorithms in Computational Geometry*. Springer-Verlag, 1987.
13. I.Couso, S.Moral, and P.Walley. A survey of concepts of independence for imprecise probabilities. *Risk, Decision and Policy*, (5):165–185, 2000.
14. I.Levi. *The Enterprise of Knowledge*. MIT Press, 1st edition, 1980.
15. J.C.F.Rocha, C.P.Campos, and F.G.Cozman. Inference in polytrees with sets of probabilities. In *19th UAI Conference*, 2003. To appear.
16. J.C.F.Rocha and F.G.Cozman. Inference with separately specified sets of probabilities in credal networks. In *18th UAI Conference*, pages 430–437. Morgan Kaufmann, 2002.
17. J.C.F.Rocha and F.G.Cozman. Inference in credal nets with branch-and-bound algorithms. In *3rd ISIPTA*, 2003. To appear.
18. J.Pearl. *Intelligent Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1st edition, 1988.
19. J.S.Ide and F.G.Cozman. Random generation of bayesian networks. In *Proc. of the XVI SBIA*, pages 366–375. Springer-Verlag, 2002.
20. L.Campos and S.Moral. Independence concepts for convex sets of probabilities. In *11th UAI Conference*, pages 108–115. Morgan Kaufmann, 1995.
21. M.Avriel. *Advances in Geometric Programming*. Plenum Press, 1980. New York.
22. M.J.Smithson. Human judgment and imprecise probabilities. Tech. Rep. citeseer.nj.nec.com/smithson97human.html, The Imprecise Probability Project, 2000.
23. M.Pradhan. The sensitivity of belief networks to imprecise probabilities: an experimental investigation. *Artificial Intelligence*, (85):363–397, 1996.
24. M.Zaffalon. *Inferenze e Decisioni in Condizioni di Incertezza con Modelli Grafici Orientati*. Ph.d. thesis, Università di Milano, Milan, Italy, 1997. in Italian.
25. P.Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, 1991.
26. P.Walley. Coherent upper and lower previsions. Tech. Rep. www.sipta.org, The Imprecise Probability Project, 2000.

Thanks to CAPES, CNPq, HP Labs and Instituto de Pesquisas Eldorado.