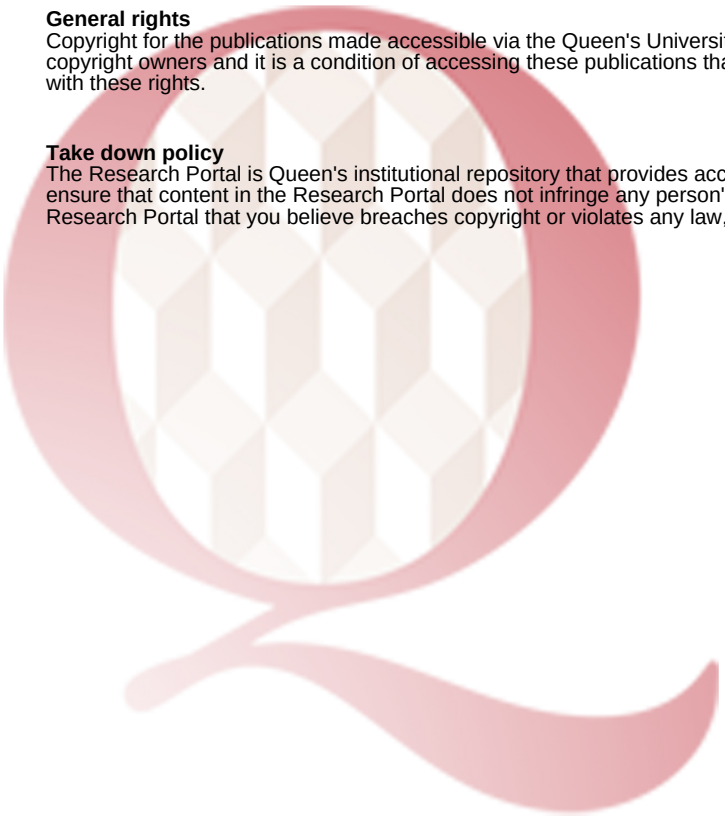
Local Computation in Credal Networks



Local Computation in Credal Networks

Fabio Gagliardi Cozman[1] and **Cassio Polpo de Campos**[2]

Abstract. The goal of this contribution is to discuss local computation in credal networks — graphical models that can represent imprecise and indeterminate probability values. We analyze the inference problem in credal networks, discuss how inference algorithms can benefit from local computation, and suggest that local computation can be particularly important in approximate inference algorithms.

1 INTRODUCTION

There are several graph-theoretic tools that simplify the representation of large multivariate models; Bayesian networks and Markov fields are two examples [24]. These two graphical models display "locality" in several dimensions: they are built from local pieces, and they are processed by local operations. These (and other) graphical models require *precise* probability assessments: the models are built in such a way that every event is associated with a single probability value.

A *credal network* is a graphical model that relaxes the assumption of uniqueness for probability values, while retaining most of the structure adopted in the Bayesian network formalism. A credal network is composed of a directed acyclic graph, a collection of random variables, and a collection of sets of probability measures — sets of probability measures are used to represent imprecision and indeterminacy in probability values. Imprecision and indeterminacy arise because beliefs may be incomplete, vague, or there may be no resources to gather/process enough information so as to reach a precise probability assessment; it may also be the case that a group of individuals must specify probability values and these individuals cannot agree on precise probability values. Section 2 presents a few basic concepts on credal networks.

Given a credal network, an *inference* is a computation of a tight lower or upper bound for some conditional probability. Unfortunately, exact inference in credal networks seems to defy strictly local approaches, as discussed in Section 3. In this contribution we analyze the inference problem in credal networks and discuss how the problem can benefit from local computations. We suggest that the main use of local information is in producing efficient approximation schemes — noting that approximate inference methods can be very useful elements of exact inference algorithms. In Section 4 we discuss the A/R++ and the MLI algorithms, two complementary and quite successful algorithms for approximate inference in credal networks.

2 CREDAL NETWORKS AND INFERENCES

Consider a few preliminary definitions. A set of probability measures is called a *credal set* [23]. A credal set defined by probability distributions $p(X)$ is denoted by $K(X)$. A *joint* credal set $K(\mathbf{X})$ contains joint probability measures for variables $\mathbf{X}$. Given a credal set $K(X)$ and an event A, the *upper* and *lower* probabilities of A are defined respectively as $\overline{P}(A) = \sup_{p(X)\in K(X)} P(A)$ and $\underline{P}(A) = \inf_{p(X)\in K(X)} P(A)$. The most commonly adopted scheme for conditioning in credal sets is elementwise Bayes rule (that is, conditioning is obtained by applying Bayes rule to each element of a credal set). Such an intuitive prescription, called the *generalized Bayes rule* by Walley [27, 28], can be justified axiomatically in various ways [19, 20, 27]. A collection of conditional credal sets $K(X|Y=y)$ is *separately specified* when the sets $K(X|Y=y_1)$ and $K(X|Y=y_2)$ are unrelated for $y_1 \neq y_2$ [14].

There are several concepts of independence that can be used when one deals with credal sets [10, 17, 27]. In this paper we adopt the concept of *strong independence*: Variables X and Y are *strongly independent* when every extreme point of the underlying credal set $K(X,Y)$ satisfies standard stochastic independence of X and Y. Similarly, X and Y are strongly independent conditional on Z when every extreme point of $K(X,Y|Z=z)$ satisfies conditional stochastic independence for all values of Z.

We can now present an appropriate definition for credal networks. We must have a directed acyclic graph, a set of random variables (one per node), and a Markov property on the graph: Every variable X is *strongly* independent of its nondescendants nonparents given its parents. Typically each variable is associated with a "local" collection of credal sets, indicated by $K(X|\mathrm{pa}(X))$ (where $\mathrm{pa}(X)$ denotes the nodes that are parents of X in the graph). Here $K(X|\mathrm{pa}(X))$ denotes a collection of credal sets, one for each value of $\mathrm{pa}(X)$. Usually these credal sets are separately specified [18, 14, 26].

Given a credal network, we consider the largest set of joint distributions that satisfy the Markov extension (for strong independence), called the *strong extension* [13]. The strong extension of a network is the convex hull of all joint distributions that satisfy the Markov property with respect to standard stochastic independence [13]. That is, given a credal network with local separately specified credal sets $K(X_i|\mathrm{pa}(X_i))$, the strong extension of the network is the convex hull of the set

$$\left\{ \prod_i p(X_i|\mathrm{pa}(X_i)) \ : p(X_i|\mathrm{pa}(X_i)=\pi_k) \in K(X_i|\mathrm{pa}(X_i)=\pi_k) \right\}. \quad (1)$$

Strong extensions were already implicit in the first proposals for credal networks [7, 26] and have received considerable attention [1, 8, 11, 18, 29]. There are other types of extension in the literature [12], but they seem to be less amenable to local computation and are not further discussed in this paper.

An *inference* in a credal network is the computation of lower/upper probabilities in an extension of the network. If X_q is a *query* variable and $\mathbf{X}_E$ represents a set of *observed* variables, then an inference is

[1] Escola Politécnica, Univ. de São Paulo, São Paulo, SP, Brazil. Email: fgcozman@usp.br
[2] PUC-SP São Paulo, SP, Brazil. Email: cassio@pucsp.br

the computation of tight bounds for $p(X_q|\mathbf{X}_E)$ for one or more values of X_q. Consider the computation of a lower probability:

$$\overline{p}(X_q|\mathbf{X}_E) = \max \frac{\sum_{X_1,\ldots,X_n \setminus X_q, \mathbf{X}_E} p(X_i|\mathrm{pa}(X_i))}{\sum_{X_1,\ldots,X_n \setminus \mathbf{X}_E} p(X_i|\mathrm{pa}(X_i))}, \quad (2)$$

$$\text{subject to } p(X_i|\mathrm{pa}(X_i)) \in K(X_i|\mathrm{pa}(X_i)) \,.$$

For inferences with strong extensions, it is known that the distributions that minimize/maximize $p(X_q|\mathbf{X}_E)$ belong to the set of vertices of the extension [18].

The only credal networks that are amenable to efficient exact inferences are polytree-shaped networks with binary variables [18]. Other types of networks, even polytree-shaped ones, face tremendous computational challenges [14]. Exact inference algorithms typically examine potential vertices of the strong extension to produce the required lower/upper values [2, 8, 12, 14, 15]. Approximate inference algorithms can produce either *outer* or *inner* approximations: the former produce intervals that enclose the correct probability interval between lower and upper probabilities [6, 21, 16, 26], while the latter produce intervals that are enclosed by the correct probability interval [1, 3, 2, 11]. Rather detailed overviews of inference algorithms for imprecise probabilities have been published by Cano and Moral [4, 5].

3 "LOCALITY" IN CREDAL NETWORKS

A credal network is clearly defined by "local" pieces of information, represented by the various local credal sets $K(X|\mathrm{pa}(X))$. We should expect that this modular structure would lead naturally to local computation in inference algorithms, much like inference in Bayesian networks. However, the picture is a little more complicated.

The only exact inference algorithm that is solely based on local computation is the 2U algorithm [18]. This algorithm deals with binary variables in polytree-shaped credal networks; its sequence of operations closely resembles Pearl's propagation scheme for Bayesian networks [24]. As in Pearl's propagation, the 2U algorithm prescribes the exchange of *messages* between variables. The 2U algorithm uses *interval-valued* messages to generate inferences. The algorithm makes critical use of the fact that a single probability interval can define any credal set for a *binary* variable.

Several other exact inference algorithms for strong extensions try to capture global information through local messages, with varying degrees of success. There are several schemes that mimic the messages in Bayesian network inference [8, 12, 14], but there is an important difference:

- Inference in Bayesian networks requires the computation of local real-valued messages that summarize probabilities in certain subnetworks. Thus a message carries local information that represents a possibly large portion of a network.
- In a strong extension, local messages are not just functions; messages are *sets* of functions. These sets of functions also summarize the credal sets in certain sub-networks. The difficulty is that a local set of functions may itself be an exceedingly complex object; in fact, a set may be as complex as the sub-network it is representing!

Thus one faces the embarassing fact that a "local" message in a credal network can be literally as rich and complex as the whole "global" content that the message is transmitting.

To emphasize the point discussed in the previous paragraph, consider the network in Figure 1. This is a very simple polytree-shaped network; suppose all variables are categorical, with four categories each. Suppose we have a number of separately specified credal sets for this network: credal sets $K(F)$, $K(B)$, $K(L|f)$ for each category of F, $K(D|f,b)$ for each category of (F,B), and $K(H|d)$ for each category of D. Finally, suppose that each one of these credal sets has exactly 3 vertices. Note that there are possibly 3^{26} different products $\prod_i p(X_i|\mathrm{pa}(X_i))$ defined by Expression (1). Depending on the characteristics of the local credal sets, all of these 3^{26} products may be vertices of the strong extension. Suppose now that H is observed and we are interested in the lower/upper probability for L conditional on H. Were this network a Bayesian network, the probability $p(L|H)$ would be produced simply by combining the local information on L, $p(L|F)$, with the local message sent by F to L, $p(F|H)$. With a credal network, we have to combine the local information on L, $K(L|F)$, with the local message sent by F to L, $K(F|H)$. The difficulty is that this local message is a set that can have up to 3^{22} vertices. Nothing is really gained, in terms of computational complexity, by the "locality" of the message $K(F|H)$.

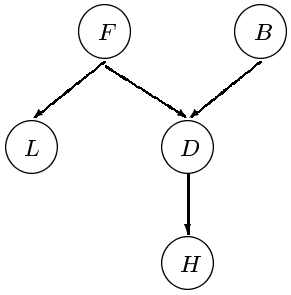


Figure 1. Example network.

It could be possible to reduce the complexity of set-valued messages in inference using redundancy elimination algorithms; that is, a message could be pre-processed before it is sent, so that its non-extreme elements are eliminated [5]. Such a strategy would conceivably lead to "local" messages that are actually less complex than the objects they represent. However, there is a non-negligible cost incurred by redundancy elimination algorithms, and there are no guarantees that the complexity reduction produced by such algorithms is significant at all — in fact existing empirical results suggest that it is not [14].

The difficulty in the way of local computation is that inference in strong extensions is essentially a global optimization problem. Consider the computation of a lower probability for variable X_q conditional on $\mathbf{X}_E$; that is, consider Expression (2). This is a non-linear optimization problem, and no set of truly local messages seems to capture the complexity of the problem.[3]

It seems difficult to construct a truly local computation scheme that finds solutions for Expression (2). Even though the constraints in Expression (2) are linear and "local" with respect to the variables in the network, the objective function defined by Expression (2) is not "local" as it contains terms from all credal sets. However, we can still explore "locality" in Expression (2). The first attempt to do so has apparently been proposed by Andersen and Hooker [1]; recently

[3] We use the term "truly local" to indicate methods that perform computation using local messages that are in fact of lower complexity than the sub-networks they are intended to summarize.

an alternative proposal has been derived by Campos and Cozman [9]. The idea of Campos and Cozman's method is to write Expression (2) as a sequence of smaller, "local" expressions, using artificial variables when necessary. To illustrate this idea, consider a simple network $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$. Assume all variables in the network are ternary. Computation of the upper probability for $\{E = e_0\}$ using Expression (2) leads to

$$\max \sum_{h,i,j,k} p(e_0|d_h)\, p(d_h|c_i)\, p(c_i|b_j)\, p(b_j|a_k)\, p(a_k)\,,$$

a multilinear function with 81 nonlinear terms of degree four. We can transform this expression by introducing new variables so as to keep the degree at most 2. We obtain just 30 nonlinear terms in $\max \sum_i p(e_0|d_i)\, p(d_i)$ subject to

$$\begin{aligned} p(d_k) &= \sum_j p(d_k|c_j)\, p(c_j)\,, \\ p(c_k) &= \sum_j p(c_k|b_j)\, p(b_j)\,, \\ p(b_k) &= \sum_j p(b_k|a_j)\, p(a_j) \end{aligned}$$

(for $k = 0, 1, 2$), plus the linear constraints. Note that these terms keep some of the "local" structure in the network. The resulting multilinear programming techniques is then solved using appropriate techniques; the most promising exact inference algorithm for credal networks is currently based on this idea [9]. Given the "global" character of the multilinear program, we do not dwell on it here. However there is an important point we want to make. When solving any multilinear program, particularly a large multilinear program, the existence of approximate solutions is critical [9]. Because the multilinear program has a structure that mirrors the structure of the underlying credal network, we can use approximate inference algorithms on the network as intermediate solutions inside the multilinear program [9]. This is precisely the advantage of using local structure to formulate the multilinear program.

Given the difficulties of local computation methods in exact inference, and the importance of approximate inference, we might ask whether local computation is a viable idea for approximate inference. Here the answer is clearly positive: not only there are many algorithms for approximate inference based on local computation, but it seems that the potential for further developments is large. In the next section we discuss two recent, and quite successful, algorithms for approximate inference based on local computation.

4 THE A/R++ AND MLI ALGORITHMS

Tessem's A/R algorithm seems to be the first local computation scheme for polytree-shaped credal networks: the algorithm essentially follows Pearl's propagation, but approximates each set-valued message with interval probabilities [26]. The A/R algorithm was later extended to general multiply connected networks, using the same types of approximations [21]. Another extension of the A/R algorithm was proposed by Rocha and Cozman [14], where the local approximations are still intervals but the combination of intervals is performed with higher precision operations. A different local approximation method is to use probability trees to represent local messages at different levels of granularity, as done by Cano and Moral [6]. All such methods produce outer approximations.

The central ideas in the A/R algorithm (and its extensions by Ha et al [21] and by Rocha et al [14]) can be cast as follows. Consider a network, a query variable X_q and evidence $\mathbf{X}_E$. We can produce lower/upper probabilities for X_q conditional on $\mathbf{X}_E$ by creating an elimination order, and by eliminating through summation one variable at a time, except X_q [12]. This is essentially the variable elimination algorithm applied to credal networks; as discussed in Section 3, here we must deal with sets of functions instead of single functions. The intermediate sets of functions generated by the method can be quite complex; the idea of the A/R algorithm algorihm is to approximate any set of functions by a single interval-valued function. This general idea is particularly elegant when the network is polytree-shaped, because then the elimination order can be replaced by Pearl's propagation scheme. The intermediate functions (the *messages*) generated in Pearl's propagation scheme are then replaced by interval-valued messages. Take for instance a node X and consider that X must send a message $\lambda_X(Y)$ to its parent Y, by combining messages received from X's other parents and children (here we follow the terminology and notation in Pearl's propagation algorithm). The A/R algorithm sends upper and lower bounds for $\lambda_X(Y)$.

The interval-valued functions that are used by the A/R algorithm (and variants) can be *easily produced by multilinear programming* — they are actually local versions of Expression (2)! Thus we obtain the following algorithm, which we call *A/R++*: eliminate variables in a credal network, but approximate the intermediate sets of functions by interval-valued functions, and compute these interval-valued functions using local multilinear programs. Here the "local" computation generates an approximation, not an exact result.

We have conducted experiments on five sets of networks, to illustrate the behaviour of inference with A/R++. Results are shown in Table 1. Each test set was composed of 10 randomly generated multiconnected credal networks (generated with BNGenerator [22]). Experiments refer to computation of upper probabilities without evidence; results refer to the most challenging inferences in each network. Table 1 indicates the topology of the test networks. Experiments were performed in a Pentium IV 1.7GHz, using Sherali and Tuncbilek's Reformulation-Linearization algorithm for multilinear programming, and CPLEX as linear solver. Further details can be found elsewhere [9].

Network topology	Type of variables	# variables	# vertices per credal set	A/R++ error
dense	binary	10	2	2.8684%
Alarm	binary	37	2	5.5706%
dense	ternary	10	3	10.4304%
Alarm	ternary	37	3	22.3293%
dense	quaternary	10	4	13.4146%

Table 1. Test sets (each with 10 networks) with average errors during inference.

There are several methods based on local computation for inner approximations. 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 $K(X_i|\mathrm{pa}(X_i))$. Methods based on gradient descent, simulated annealing and genetic programming pursue this idea [1, 2, 3, 11, 29].

A particularly successful scheme is the MLI algorithm presented by Rocha et al [16]. The MLI algorithm tries to use the fact that the non-linear problem (2) has linear constraints that are "local" to their associated credal sets. The algorithm fixes a vertex for every credal set except one, and checks which vertex of the remaining credal set minimizes/maximizes $p(X_q|\mathbf{X}_E)$ (given that all the others are fixed).

The algorithm then retains the minimizing/maximizing vertex, and then move to the next credal set. Now all the vertices are fixed, except for this next credal set, *using the minimizing/maximizing* vertex obtained in the previous step. The algorithm keeps repeating these steps, going over and over all the local credal sets in the credal network. The process is surely to stop: every step increases the objective function, and there is only a finite number of possible moves (given that variables are discrete and local credal sets have finitely many vertices). Implementation details can be found in [16].

The MLI algorithm typically produces very accurate approximations. We have run it in a large number of medium-sized networks, and verified that in most cases it finds the exact answer, and always finds a very accurate approximation.

5 CONCLUSION

We have tried to provide a brief but coherent commentary of local computation in credal networks. These graphical models are interesting tools for representation of several forms of uncertainty, and they have a modular (and therefore "local") structure. In short, local computation is quite important in approximate algorithms, and less directly applicable in exact algorithms. We should stress that any approximate inference algorithm can be used inside other exact inference algorithms: As the exact solution of Expression (2) usually employs branch-and-bound (or similar) techniques, any guiding approximation can speed up exact inference in very significant ways [9].

It seems that several local computation techniques could be applied to approximate inference in the near future, with a potential for excellent results. The use of local computation in exact inference is also promising but remains a challenge for the most part.

Acknowledgements

This work was (partially) developed in collaboration with HP Brazil R&D. The first author has also been partially supported by CNPq (through grant 3000183/98-4).

REFERENCES

[1] K. A. Andersen and J. N. Hooker. Bayesian logic. *Decision Support Systems*, 11:191–210, 1994.

[2] A. Cano, J. E. Cano, and S. Moral. Convex sets of probabilities propagation by simulated annealing. In G. Goos, J. Hartmanis, and J. van Leeuwen, editors, *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 4–8, Paris, France, July 1994.

[3] A. Cano and S. Moral. A genetic algorithm to approximate convex sets of probabilities. *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2:859–864, 1996.

[4] A. Cano and S. Moral. A review of propagation algorithms for imprecise probabilities. In G. de Cooman, F. G. Cozman, S. Moral, and P. Walley, editors, *Int. Symp. on Imprecise Probabilities and Their Applications*, pages 51–60, Ghent, Belgium, 1999. Imprecise Probabilities Project, Universiteit Ghent.

[5] A. Cano and S. Moral. Algorithms for imprecise probabilities. In *Handbook of Defeasible and Uncertainty Management Systems*, pages 369–420, Kluwer Academic Publishers, 2000.

[6] A. Cano and S. Moral. Using probability trees to compute marginals with imprecise probabilities. *Int. Journal of Approximate Reasoning*, 29:1–46, 2002.

[7] J. Cano, M. Delgado, and S. Moral. Propagation of uncertainty in dependence graphs. *European Conf. on Symbolic and Quantitative Approaches to Uncertainty*, pages 42–47, 1991.

[8] J. Cano, M. Delgado, and S. Moral. An axiomatic framework for propagating uncertainty in directed acyclic networks. *Int. Journal of Approximate Reasoning*, 8:253–280, 1993.

[9] C. P. de Campos, F. G. Cozman. Inference in credal networks using multilinear programming. STAIRS, 2004.

[10] I. Couso, S. Moral, and P. Walley. A survey of concepts of independence for imprecise probabilities. *Risk, Decision and Policy*, 5:165–181, 2000.

[11] F. G. Cozman. Robustness analysis of Bayesian networks with local convex sets of distributions. In D. Geiger and P. Shenoy, editors, *Conf. on Uncertainty in Artificial Intelligence*, pages 108–115, San Francisco, 1997. Morgan Kaufmann.

[12] F. G. Cozman. Credal networks. *Artificial Intelligence*, 120:199–233, 2000.

[13] F. G. Cozman. Separation properties of sets of probabilities. In C. Boutilier and M. Goldszmidt, editors, *Conf. on Uncertainty in Artificial Intelligence*, pages 107–115, San Francisco, July 2000. Morgan Kaufmann.

[14] J. C. F. da Rocha and F. G. Cozman. Inference with separately specified sets of probabilities in credal networks. In A. Darwiche and N. Friedman, editors, *Proceedings of the Conf. on Uncertainty in Artificial Intelligence*, pages 430–437, San Francisco, California, 2002. Morgan Kaufmann.

[15] J. C. F. da Rocha and F. G. Cozman. Inference in credal networks with branch-and-bound algorithms. In *Int. Symp. on Imprecise Probabilities and Their Applications*, pages 482–495. Carleton Scientific, 2003.

[16] J. C. F. da Rocha, F. G. Cozman, and C. P. de Campos. Inference in polytrees with sets of probabilities. In *Conf. on Uncertainty in Artificial Intelligence*, pages 217–224, San Francisco, California, 2003. Morgan Kaufmann.

[17] L. de Campos and S. Moral. Independence concepts for convex sets of probabilities. In P. Besnard and S. Hanks, editors, *Conf. on Uncertainty in Artificial Intelligence*, pages 108–115, San Francisco, California, 1995. Morgan Kaufmann.

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

[19] F. J. Giron and S. Rios. Quasi-Bayesian behaviour: A more realistic approach to decision making? In J. M. Bernardo, J. H. DeGroot, D. V. Lindley, and A. F. M. Smith, editors, *Bayesian Statistics*, pages 17–38. University Press, Valencia, Spain, 1980.

[20] A. J. Grove and J. Y. Halpern. Updating sets of probabilities. In *Conf. on Uncertainty in Artificial Intelligence*, pages 173–182, San Francisco, California, 2002. Morgan Kaufmann.

[21] V. A. Ha, A. Doan, V. H. Vu, and P. Haddawy. Geometric foundations for interval-based probabilities. *Annals of Mathematics and Artificial Intelligence*, 24(1-4):1–21, 1998.

[22] J. S. Ide and F. G. Cozman. Random generation of Bayesian networks. In *Brazilian Symp. on Artificial Intelligence (SBIA)*, pages 366–375, Porto de Galinhas, Pernambuco, Brasil, 2002.

[23] I. Levi. *The Enterprise of Knowledge*. MIT Press, Cambridge, Massachusetts, 1980.

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

[25] H. D. Sherali and C. H. Tuncbilek. A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. *Journal of Global Optimization*, 2:101–112, 1992.

[26] B. Tessem. Interval probability propagation. *Int. Journal of Approximate Reasoning*, 7:95–120, 1992.

[27] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.

[28] P. Walley. Measures of uncertainty in expert systems. *Artificial Intelligence*, 83:1–58, 1996.

[29] M. Zaffalon. *Inferenze e Decisioni in Condizioni di Incertezza con Modelli Grafici Orientati*. PhD thesis, Università di Milano, February 1997.