Computing Lower and Upper Expectations under Epistemic Independence

Cassio Polpo de Campos
PUC-SP,
São Paulo, SP, Brazil
email: cassio@pucsp.br

Fabio G. Cozman
Escola Politécnica, Univ. de São Paulo,
São Paulo, SP, Brazil
email: fgcozman@usp.br

Abstract

This papers investigates the computation of lower/upper expectations that must cohere with a collection of probabilistic assessments and a collection of judgements of epistemic independence. New algorithms, based on multilinear programming, are presented, both for independence among events and among random variables. Separation properties of graphical models are also investigated.

1 Introduction

Among the concepts of independence that have been investigated in connection with sets of probability measures, the concept of *epistemic irrelevance* is probably the easiest to explain — intuitively, Y is epistemically irrelevant to X if assessments about X do not change when we observe Y [33]. *Epistemic independence* is the symmetric concept: X and Y are epistemically independent if each one is epistemically irrelevant to the other. Despite their intuitive content, epistemic irrelevance and epistemic independence are quite difficult to handle computationally. Given probabilistic assessments and judgements of epistemic irrelevance, how can one compute lower and upper expectations?

Our main contribution in this paper is to show that judgements of epistemic irrelevance can generally be recast as multilinear constraints. We show how to compute lower/upper expectations that take into account epistemic irrelevance through multilinear programming. We apply our multilinear approach to multivariate models with graph-theoretical representations, often called credal networks. We consider credal networks under "epistemic irrelevance" and "epistemic independence" semantics, and investigate separation properties of these networks.

Section 2 presents a few relevant definitions and results. Section 3 introduces our multilinear approach to epistemic irrelevance among events (Appendix A compares our approach to Walley's algorithm for epistemic irrelevance). Sections 4 and 5 look respectively into credal networks and separation properties of Markov chains. Section 6 concludes the paper.

2 Credal sets and concepts of independence

We deal with *categorical* random variables. To distinguish random variables from (non-random) variables employed in optimization problems, we refer to the latter as *optimization variables*.

A set of probability measures induced by distributions on random variable X is denoted by $K(X)$ and called a *credal set*. A *joint* credal set $K(\mathbf{X})$ contains joint probability measures for random variables $\mathbf{X}$. Conditioning is performed by applying Bayes rule to each measure in a credal set; the posterior credal set is the union of all posterior probability measures [17]. A *conditional* credal set $K(X|A)$ contains conditional measures on the event A. Given a credal set $K(X)$, the *lower expectation* and the *upper expectation* of a bounded function $f(X)$ are defined respectively as $\underline{E}[f(X)] = \inf_{p(X)\in K(X)} E[f(X)]$ and $\overline{E}[f(X)] = \sup_{p(X)\in K(X)} E[f(X)]$, where $E[f(X)]$ is standard expectation. Lower/upper probabilities are defined similarly, as are conditional lower/upper expectations/probabilities.

We defer to future work the very important case of conditioning on events with zero probability [4, 9, 31]; here we assume throughout that any conditioning event has lower probability larger than zero.

Lower and upper expectations can be viewed as linear constraints on probabilities: $\underline{E}[f(X)] \leq E_P[f(X)] \leq \overline{E}[f(X)]$. Conditional lower and upper expectations also yield linear constraints, as $\underline{E}[f(X)|A] = \alpha$ is equivalent to $\underline{E}[A(X)(f(X) - \alpha)] = 0$, where we use $A(X)$ for the indicator function of event A (this equation is Walley's *generalized Bayes rule* [33]). If a collection of lower/upper expectations defines a convex set of probability measures, such that every constraint is tight, we say that the lower/upper expectations are *coherent*. We do not assume that every given set of constraints is coherent; we as-

sume only that any set of constraints defines a non-empty set of measures and thus can be made coherent by adjusting some assessments. A set of constraints with this property is said to *avoid sure loss* [33].

In general, we are interested in the *largest* set of probability measures that satisfies a given set of constraints — these constraints may be coherent or not, but they must avoid sure loss. We call this largest set the *natural extension* of the constraints, borrowing from Walley's terminology [33].

Several concepts of independence can be used when one deals with credal sets [5, 10, 14, 33]. We review here three non-equivalent concepts; relationships between them have received considerable attention in the literature [8, 11, 24].

The most commonly adopted concept is *strong independence*:[1] Events A and B are *strongly independent* when every extreme point of the underlying credal set K satisfies standard stochastic independence of A and B. Likewise, random variables X and Y are strongly independent when every extreme point of the underlying credal set satisfies standard stochastic independence of X and Y. Conditional strong independence (for events and for random variables) is obtained by demanding that extreme points satisfy stochastic independence conditional on a given event. Strong independence usually produces a multilinear program, as the following example illustrates.[2]

Example 1 Consider a generalized version of Boole's challenge problem [19]. Take three Boolean random variables X_1, X_2 and X_3; random variable X_i takes values i and $\hat{i}$. We want to find tight bounds on $P(X_3 = 3)$. Whenever possible we indicate the events $\{X_i = i\}$ and $\{X_i = \hat{i}\}$ simply by i and $\hat{i}$, and we indicate conjunction of events $A \wedge B$ simply by A, B. Suppose we have $P(1) \in [l_1, u_1]$, $P(2) \in [l_2, u_2]$, $P(1,3) \in [l_3, u_3]$, $P(2,3) \in [l_4, u_4]$, $P(\hat{1}, \hat{2}, 3) = 0$, with $l_i > 0$. Suppose also that X_1 and X_2 are strongly independent; given that relevant probabilities are positive, strong independence implies $P(1,2) = P(1)P(2)$ for every vertex of $K(X_1, X_2)$. Defining $p_1 = P(1,2,3)$, $p_2 = P(1,2,\hat{3})$, $p_3 = P(1,\hat{2},3)$, $p_4 = P(1,\hat{2},\hat{3})$, $p_5 = P(\hat{1},2,3)$, $p_6 = P(\hat{1},2,\hat{3})$, $p_7 = P(\hat{1},\hat{2},3)$, $p_8 = P(\hat{1},\hat{2},\hat{3})$, we have:

$$\begin{aligned}
&\max / \min\ p_1 + p_3 + p_5 + p_7, \quad \text{subject to} \\
&p_1 + p_2 + p_3 + p_4 = \pi_1, \quad p_1 + p_2 + p_5 + p_6 = \pi_2, \\
&p_1 + p_3 = \pi_3, \quad p_1 + p_5 = \pi_4, \quad p_1 + p_2 = \pi_1\pi_2, \\
&p_7 = 0, p_1 + \cdots + p_8 = 1, l_i \leq \pi_i \leq u_i, p_k \geq 0.
\end{aligned}$$

Suppose that $l_1 = 0.1$, $l_2 = 0.2$, $l_3 = 0.1$, $l_4 = 0.3$, $u_1 = 0.5$, $u_2 = 0.8$, $u_3 = 0.3$, and $u_4 = 0.7$. The solution of this multilinear program yields $P(3) \in [0.3, 0.79]$. If the independence judgement is dropped, then linear programming produces $P(3) \in [0.3, 1.0]$. □

Unlike geometric programs, multilinear constraints lead to nonconvex primal and dual programs, and no known transformation can convexify them. Existing solution methods produce sequences of sub-problems using either branch-and-bound or cutting-plane techniques [18, 20, 22, 26, 29]. The algorithms of Maranas and Floudas [22], and Gochet and Smeers [18] produce convex nonlinear sub-problems, while Sherali and Adams' algorithm produces linear sub-problems [26]. We employ Sherali and Adams' branch-and-bound algorithm in our calculations, as it is particularly appropriate for computing lower/upper expectations — because the sub-problems generated by this method are linear programs, column generation and other valuable techniques can be employed [19].

A different definition of independence is Kuznetsov's: X and Y are *Kuznetsov independent* when the interval of expected values $\mathbb{E}[f(X)g(Y)]$ is equal to the interval-product of the intervals $\mathbb{E}[f(X)]$ and $\mathbb{E}[g(Y)]$, for any bounded $f(X)$ and $g(Y)$ [21]. Little is known about the computation of lower/upper expectations under judgements of Kuznetsov independence; the available method works by explicitly constructing a joint credal set [10], a potentially complex operation that is not applicable to large multivariate settings in any obvious way.

A third concept of independence for credal sets is *epistemic independence* [32, 33]. In many ways, this is the concept with the most appealing definition, because it can be given a direct behavioral interpretation. We now present the relevant definitions both for events and random variables:

Definition 1 *Event A is* epistemically irrelevant *to event B given event C when $\underline{P}(B|A,C) = \underline{P}(B|A^{\complement},C) = \underline{P}(B|C)$ and $\overline{P}(B|A,C) = \overline{P}(B|A^{\complement},C) = \overline{P}(B|C)$.*

We indicate that A is epistemically irrelevant to B given C by $EIR(A,B|C)$.

Definition 2 *Events A and B are* epistemically independent *given event C when $EIR(A,B|C)$ and $EIR(B,A|C)$.*

Definition 3 *Random variable X is* epistemically irrelevant *to random variable Y given event C when $\underline{E}[f(Y)|X=x,C] = \underline{E}[f(Y)|C]$ for any bounded $f(Y)$ and any x.*

We indicate that X is epistemically irrelevant to Y given C by $EIR(X,Y|C)$.

Definition 4 *Random variables X and Y are* epistemically independent *given event C when $EIR(X,Y|C)$ and $EIR(Y,X|C)$.*

[1] We should note that terminology is not completely standardized on this topic [5, 8].

[2] Multilinear programming has also been related to other concepts of independence, for example independence in comparative probabilities [3].

We indicate that A and B are epistemically independent given C by $EIN(A, B|C)$. Likewise, $EIN(X, Y|C)$ indicates that X and Y are epistemically independent given C. We can also have irrelevance and independence conditional on a random variable Z; as we restrict ourselves to categorical random variables, the judgement $EIR(X, Y|Z)$ simply means that $EIR(X, Y|Z = z)$ for every value z of Z (and likewise for epistemic independence).

3 Epistemic independence for events

In this section we propose a multilinear programming formulation for the computation of upper expectations under judgements of epistemic irrelevance of events. The computation of lower expectations can be tackled with the same methods. We focus on epistemic irrelevance as any judgement of epistemic independence can be expressed as two judgements of epistemic irrelevance.

Consider that s assessments are given as $\underline{P}(F_i|G_i) = \alpha_i$; assessments are not necessarily coherent. Suppose we want to compute $\overline{P}(D)$ for an event D. Suppose we have N atomic events — each atomic event is a conjunction of events involved in assessments. Note that N can be exponential on the number of assessments and judgements. Denote by p_k the probability of the kth atomic event. The probability of any event A can be written as $\sum_k A_k p_k$, where A_k is the indicator function of A. Every assessment $P(F_i|G_i) \geq \alpha_i$ can be encoded as

$$E[G_i(X)(F_i(X) - \alpha_i)] \geq 0, \tag{1}$$

where $G_i(X)$ and $F_i(X)$ denote indicator functions (if the ith assessment is unconditional, $G_i(X) = 1$ for every X). From now on we drop the argument X whenever possible inside expectations.

Hence we can write $\overline{P}(D)$ as $\max \sum_k D_k p_k$, where D_k is the indicator function of event D, subject to the linear constraints $P(F_i|G_i) \geq \alpha_i$ (also expressed in terms of the p_k). Note that we are only enforcing $\underline{E}[F_i|G_i] \geq \alpha_i$, not that $\underline{E}[F_i|G_i] = \alpha_i$; if the assessments are not coherent, it may be impossible to enforce equality. Thus the flexibility of Expression (1) seems appropriate in practice.

At this point we have encoded assessments (conditional or not) into a linear program, as usually done in probabilistic logic [19] — note that the "variables" of the linear program are the atomic probabilities p_k. We emphasize: to avoid any confusion between random variables and these "variables" we refer to the latter as optimization variables.

Now consider that r judgements of epistemic irrelevance are given as $EIR(A_j, B_j|C_j)$. These judgements are harder to express, as each $EIR(A_j, B_j|C_j)$ introduces constraints such as $\min P(B_j|A_j, C_j) = \min P(B_j|C_j)$, where both minima are taken with respect to the underlying credal set. As we now show, it is possible to express irrelevance relations through multilinear constraints. To do so, introduce new optimization variables ν_j and μ_j, and generate the following inequalities (note that inequality symbols are numbered, as their order is used later):

$$\begin{array}{lllll} \nu_j & \leq_1 & P(B_j|A_j, C_j) & \leq_4 & \mu_j, \\ \nu_j & \leq_2 & P(B_j|A_j^{\complement}, C_j) & \leq_5 & \mu_j, \\ \nu_j & \leq_3 & P(B_j|C_j) & \leq_6 & \mu_j. \end{array} \tag{2}$$

By clearing the denominators, these inequalities become multilinear expressions on the p_k, ν_j and μ_j. Note that we can clear the denominators given our assumption of positive conditioning events.

Denote by $\mathcal{C}_0$ the set of assessments $\underline{E}[G_i(F_i - \alpha_i)] \geq 0$, plus the constraints $p_k \geq 0$ and the $6r$ inequalities (2). Now construct $6r$ additional sets of N optimization variables. Denote by $\mathbf{q}_{j,l}$ each one of these $6r$ sets of optimization variables — there is one set for each judgement of irrelevance (where $j = 1, \ldots, r$) and for each inequality in (2) (where $l = 1, \ldots, 6$ indicates which inequality is used, following the numbering in (2)).

The idea is simple. For each judgement of irrelevance and each inequality, there must be a measure on the underlying joint credal that satisfies the inequality with equality. As each inequality may be satisfied with equality by a different measure, we must create as many measures as there are inequalities. For example, optimization variables $\mathbf{q}_{3,4}$ will have to satisfy $P(B_3|A_3, C_3) = \mu_3$, or rather

$$P(A_3, B_3, C_3) = \mu_3 P(A_3, C_3). \tag{3}$$

Thus we construct $6r$ sets of constraints. The set of constraints $\mathcal{C}_{j,l}$ only refers to optimization variables in $\mathbf{q}_{j,l}$. The constraints are identical to the ones in $\mathcal{C}_0$, except that: (1) instead of optimization variable p_k we have $q_{j,l,k}$; (2) the lth inequality is replaced by equality. We obtain a set of $6r + 1$ loosely coupled systems of multilinear constraints; the connection between these systems is given by the ν_j and μ_j. By construction, we have:

Theorem 1 *The value of $\overline{P}(D)$ is given by $\max P(D)$ (as a linear expression of p_k) subject to $\mathcal{C}_0$, $\sum_k p_k = 1$, $\mathcal{C}_{j,l}$, and $\sum_k q_{j,l,k} = 1$ for $j = 1, \ldots, r$ and $l = 1, \ldots, 6$.*

To illustrate this result, we revisit Example 1:

Example 2 Consider the same assessments described in Example 1, but replace the strong independence judgement with the epistemic independence judgement $EIN(1, 2)$. To compute $P(3)$ we must deal with 13 groups of 8 optimization variables and approximately 300 constraints, many of which are multilinear. Our implementation of Sherali and Adams' method readily produces $P(3) \in [0.3, 0.85]$. □

The previous discussion can be adapted to produce conditional upper expectations of the form $\overline{P}(D|E)$. We start with a fractional multilinear program where the objective function is $\max P(D,E)/P(E)$. Now define $t = P(E)$; the objective function then is $\max t^{-1} \sum_k D_k E_k p_k$. Given our assumption that $t > 0$, we can multiply by t^{-1} both sides of constraints (1), (2) or (3). If we distribute t^{-1} and replace every product $t^{-1}p_k$ by a new optimization variable p'_k, and every product $t^{-1}q_{j,l,k}$ by a new optimization variable $q'_{j,l,k}$, we obtain a multilinear program that is essentially identical to the original fractional multilinear program. There are a few differences; most notably, the objective function becomes $\max \sum_k D_k E_k p'_k$, where E_k denotes the indicator function of E. Also, the definition $t = P(E)$ leads to the constraint $\sum_k E_k p'_k = 1$. Finally, the unitary constraint $\sum_k p_k = 1$ becomes $\sum_k p'_k = t^{-1}$, and in fact this is the only constraint that contains t — thus it can be suppressed in the presence of the other constraints. Note that this technique mimics the Charnes-Cooper transformation used in linear fractional programming [6].

The techniques outlined in this section remain essentially untouched if we consider assessments with functions of random variables such as $\underline{E}[f_i(X)|G_i(X)] = \alpha_i$; we must then handle constraints $E[G_i(X)(f_i(X) - \alpha_i)] \geq 0$. Note again that we translate the assessments into inequalities (not equalities) as we admit assessements that may not be coherent.

Section 6 briefly compares our multilinear programming approach with Walley's iterative algorithm (presented in Appendix A).

4 Epistemic independence for random variables: credal networks

While judgements of epistemic independence between events imply a fixed number of equalities among lower and upper probabilities, epistemic independence between random variables requires that credal sets have identical convex hulls — and these convex hulls can be rather complex objects. In Appendix A we derive a generalization of Walley's algorithm that deals with arbitrary judgments of independence between random variables, but the resulting method faces steep computational difficulties. Instead of dealing with arbitrary judgements of independence, in this section we focus on judgements that can be organized using graph-theoretical tools. We explore compact representations for credal sets that are inspired by Bayesian networks and other graphical models [25].

We thus consider *credal networks* as our representation for judgements of epistemic irrelevance and independence [1, 2, 8, 15]. A credal network consists of a directed acyclic graph where each node is associated with a random variable X_i and with *local credal sets*. The local credal sets for X_i contains probability measures for random variable X_i conditional on the values of random variables that are *parents* of X_i in the directed acyclic graph. We denote parents of X_i by $\mathrm{pa}(X_i)$ and local credal sets by $K(X_i|\mathrm{pa}(X_i))$. We assume that local credal sets are *separately specified*, that is, $K(X_i|\mathrm{pa}(X_i) = \pi_i)$ and $K(X_i|\mathrm{pa}(X_i) = \pi_j)$ impose no constraints on each other for $\pi_i \neq \pi_j$.

Here we are interested in semantics for credal networks that are based on epistemic irrelevance; we thus consider two possible interpretations for a credal network [7]:

- The *epistemic extension based on irrelevance* is the largest joint credal set such that nondescendants nonparents of a random variable X_i are epistemically irrelevant to X_i given the parents of X_i.
- The *epistemic extension based on independence*, or simply *epistemic extension*, is the largest joint credal set such that nondescendants nonparents of a random variable X_i are epistemically independent of X_i given the parents of X_i.

These extensions are clearly based on different Markov conditions.

Suppose a credal network is given and we must compute the upper probability $\overline{P}(Q|E)$, where Q and E denote events defined by (possibly several) X_i. For the epistemic extension based on irrelevance, this computation can be reduced to a linear program [7]. To understand this reduction, consider the judgement:

$$K(X_i|\mathrm{pa}(X_i), Y_i) \cong K(X_i|\mathrm{pa}(X_i))\,, \qquad (4)$$

where Y_i represents the nondescendants nonparents of X_i, and the symbol $\cong$ indicates that credal sets must have identical convex hulls. The right hand side of expression (4) is known, as it is part of the network definition. So we can express constraints in the epistemic extension based on irrelevance by taking the constraints over $K(X_i|\mathrm{pa}(X_i))$ and replicating them for all sets $K(X_i|\mathrm{pa}(X_i), Y_i = y_i)$, for every value y_i. Constraints must be expressed over p_k, the probabilities of atomic events; as the number of atomic events is exponential on the number of random variables X_i, we obtain a potentially large linear program.

Handling epistemic extensions based on independence raises more difficulties. Such extensions must satisfy constraints (4) and the "backward" judgements

$$K(Y_i|\mathrm{pa}(X_i), X_i) \cong K(Y_i|\mathrm{pa}(X_i))\,, \qquad (5)$$

where again we denote the nondescendants nonparents of X_i by Y_i. Neither side of these constraints is directly specified by the network. This difficulty is circumvented in a "brute-force" manner by the only existing algorithm for

MULTILINEAREXTENSION($\mathbf{X}$,p_k)

$\mathbf{X}$ is a set of random variables X_i that constitute a network,

p_k is a set of variables representing atomic probabilities over $\mathbf{X}$.

(1) Generate $p_k \geq 0$ for all k and $\sum_k p_k = 1$.

(2) For every "forward" irrelevance judgement (4), generate constraints that enforce $P(X_i|\mathrm{pa}(X_i), Y_i) \in K(X_i|\mathrm{pa}(X_i))$ for every value of Y_i, using the constraints for $K(X_i|\mathrm{pa}(X_i))$ in the network description.

(3) For every random variable X_i, and for every value x_{ij}:

(3.1) Introduce variables $q_{ij}(Y_i, \mathrm{pa}(X_i))$, indexed by $\{Y_i, \mathrm{pa}(X_i)\}$, and generate constraints (one per value of $\{Y_i, \mathrm{pa}(X_i)\}$)

$$q_{ij}(Y_i, \mathrm{pa}(X_i)) \times P(\mathrm{pa}(X_i), X_i = x_{ij}) = P(Y_i, \mathrm{pa}(X_i), X_i = x_{ij}) \times \sum_{Y_i} q_{ij}(Y_i, \mathrm{pa}(X_i)).$$

(3.2) Recursively call MULTILINEAREXTENSION($\{Y_i, \mathrm{pa}(X_i)\}$, $q_{ij}(Y_i, \mathrm{pa}(X_i))$) if the network represented by $\{Y_i, \mathrm{pa}(X_i)\}$ has more than one node and contains irrelevance relations; otherwise just impose the (linear) constraints on this network over $q_{ij}(Y_i, \mathrm{pa}(X_i))$.

Figure 1: The procedure MULTILINEAREXTENSION.

epistemic extensions [7], which we call the E^3 algorithm (for *Extensive Epistemic Extension* algorithm). This algorithm explicitly builds each set appearing on the right hand side of expression (5). This construction is exponential on the number of variables; even worse, the number of constraints grows extremely fast as it requires exponentially many projections of polyhedra (each one of which with worst-case exponential complexity). Such complexity level has prevented networks with more than four variables to be dealt with in practice. Alas, the E^3 algorithm offers no clear path to approximation schemes — a frustrating situation as it seems that approximation algorithms are a necessary route to follow.

In the remainder of this section we offer a multilinear programming formulation for epistemic extensions. The algorithm we derive is significantly simpler to implement than the E^3 algorithm, and it does not require an explicit construction of the epistemic extension.

Given a credal network, we start by defining the optimization variables p_k as in Section 2; that is, these optimization variables represent atomic probabilities. We now formulate the question: what are the constraints over p_k such that these optimization variables do represent a measure in the epistemic extension? Clearly we must have $p_k \geq 0$ for all p_k, the unitary constraint $\sum_k p_k = 1$, and the "forward" judgements of irrelevance in Expression (4). These latter constraints can be written following the replication technique already discussed.

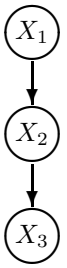
Consider now a "backward" constraint (5). We must guarantee that $P(Y_i|\mathrm{pa}(X_i), X_i = x_{ij})$ belongs to $K(Y_i|\mathrm{pa}(X_i))$ for each value x_{ij}. First we treat each value of $P(Y_i|\mathrm{pa}(X_i), X_i = x_{ij})$ as an optimization variable that is related to the p_k through the multilinear constraint $P(Y_i|\mathrm{pa}(X_i), X_i = x_{ij}) \times P(\mathrm{pa}(X_i), X_i = x_{ij}) = P(Y_i, \mathrm{pa}(X_i), X_i = x_{ij})$. Note that $P(Y_i|\mathrm{pa}(X_i), X_i = x_{ij})$ stands for optimization variables indexed by Y_i and $\mathrm{pa}(X_i)$, while $P(\mathrm{pa}(X_i), X_i = x_{ij})$ and $P(Y_i, \mathrm{pa}(X_i), X_i = x_{ij})$ are not optimization variables; they simply stand for linear functions of the optimization variables p_k. Our next step is to introduce optimization variables $q_{ij}(Y_i, \mathrm{pa}(X_i))$ that represent a "fresh" measure over $\{Y_i, \mathrm{pa}(X_i)\}$; these variables are again indexed by Y_i and $\mathrm{pa}(X_i)$. The "backward" constraint (5) requires exactly that there must be a marginal measure over $\{Y_i, \mathrm{pa}(X_i)\}$ such that $P(Y_i|\mathrm{pa}(X_i), X_i = x_{ij})$ plays the role of a distribution for Y_i conditional on $\mathrm{pa}(X_i)$. Thus we introduce the multilinear constraint

$$q_{ij}(Y_i, \mathrm{pa}(X_i)) = P(Y_i|\mathrm{pa}(X_i), X_i = x_{ij}) \times \sum_{Y_i} q_{ij}(Y_i, \mathrm{pa}(X_i)).$$

The remaining problem is to constrain the optimization variables $q_{ij}(Y_i, \mathrm{pa}(X_i))$ so that they represent a valid marginal measure over $\{Y_i, \mathrm{pa}(X_i)\}$.

At this point it is reasonable to pause and try to build some additional intuition for the status of the optimization variables $q_{ij}(Y_i, \mathrm{pa}(X_i))$. There is one such optimization variable for each combination of values of $\{Y_i, \mathrm{pa}(X_i)\}$. This set of optimization variables is introduced to guarantee that, for a given set of optimization variables $P(Y_i|\mathrm{pa}(X_i), X_i = x_{ij})$ that are part of the solution, we have a valid distribution in the set $K(Y_i|\mathrm{pa}(X_i))$ — this distribution is exactly represented by the optimization variables $q_{ij}(Y_i|\mathrm{pa}(X_i))$. Now the challenge is to guarantee that $q_{ij}(Y_i|\mathrm{pa}(X_i))$ in fact represent a distribution in $K(Y_i|\mathrm{pa}(X_i))$.

To proceed, we must note that $\{Y_i, \mathrm{pa}(X_i)\}$ form a *top sub-network* — that is, a sub-network such that if W_i is in



(a)

$p_k \geq 0$ for all k
$\sum_k p_k = 1$
$\underline{P}(1) \leq P(1) \leq \overline{P}(1)$
$\underline{P}(2|1) \leq P(2|1) \leq \overline{P}(2|1)$
$\underline{P}(2|\hat{1}) \leq P(2|\hat{1}) \leq \overline{P}(2|\hat{1})$
$\underline{P}(3|2) \leq P(3|2,1) \leq \overline{P}(3|2)$
$\underline{P}(3|\hat{2}) \leq P(3|\hat{2},1) \leq \overline{P}(3|\hat{2})$
$\underline{P}(3|2) \leq P(3|2,\hat{1}) \leq \overline{P}(3|2)$
$\underline{P}(3|\hat{2}) \leq P(3|\hat{2},\hat{1}) \leq \overline{P}(3|\hat{2})$

(b)

$$P(2,3) \times q_{3,3}(1,2) = P(1,2,3) \times (q_{3,3}(1,2) + q_{3,3}(\hat{1},2))$$
$$P(\hat{2},3) \times q_{3,3}(1,\hat{2}) = P(1,\hat{2},3) \times (q_{3,3}(1,\hat{2}) + q_{3,3}(\hat{1},\hat{2}))$$
$$P(2,3) \times q_{3,3}(\hat{1},2) = P(\hat{1},2,3) \times (q_{3,3}(1,2) + q_{3,3}(\hat{1},2))$$
$$P(\hat{2},3) \times q_{3,3}(\hat{1},\hat{2}) = P(\hat{1},\hat{2},3) \times (q_{3,3}(1,\hat{2}) + q_{3,3}(\hat{1},\hat{2}))$$

(c)

$q_{3,3}(X_1, X_2) \geq 0$, all X_1, X_2
$\sum_{X_1,X_2} q_{3,3}(X_1, X_2) = 1, \quad \underline{P}(1) \leq q_{3,3}(1) \leq \overline{P}(1)$
$\underline{P}(2|1) \leq q_{3,3}(2|1) \leq \overline{P}(2|1), \quad \underline{P}(2|\hat{1}) \leq q_{3,3}(2|\hat{1}) \leq \overline{P}(2|\hat{1})$

(d)

Figure 2: Example 3.

the sub-network then all ascendants of W_i are in the sub-network. We now use the following property [7]: the natural extension of a top sub-network, taking into account independence relations in the top sub-network, is always equal to the marginal credal set obtained by marginalizing the complete epistemic extension. That is, if we "cut" a top sub-network out of a credal network, and compute the epistemic extension for this sub-network, we obtain the same credal set we would obtain if we started with the whole network and then marginalized the whole epistemic extension. Consequently, we can force $q_{ij}(Y_i, \mathrm{pa}(X_i))$ to be a valid marginal measure by recursively calling the algorithm on the top sub-network with $\{Y_i, \mathrm{pa}(X_i)\}$. Note that no recursive call is needed when a network with a single node is processed (or a network with no independence relation). Each recursive call is applied to a smaller network; thus the procedure must terminate. The whole algorithm is described in Figure 1.

Example 3 Consider a Markov chain with three binary random variables X_1, X_2 and X_3 (Figure 2.a). As in Example 1, random variable X_i takes values i and $\hat{i}$. Suppose we have separately specified sets $K(X_1)$ (specified by $\underline{P}(1)$ and $\overline{P}(1)$), $K(X_2|X_1)$ (specified by $\underline{P}(2|1)$, $\overline{P}(2|1)$, $\underline{P}(2|\hat{1})$, $\overline{P}(2|\hat{1})$), and $K(X_3|X_2)$ (specified by $\underline{P}(3|2)$, $\overline{P}(3|2)$, $\underline{P}(3|\hat{2})$, $\overline{P}(3|\hat{2})$). The epistemic extension must satisfy $EIN(X_1, X_3|X_2)$. We have 8 variables p_k defined as in Example 1. Figure 2.b shows some basic constraints on p_k, implied directly by the local credal sets and the "forward" irrelevance judgement $EIR(X_1, X_3|X_2)$. To satisfy the judgement $EIR(X_3, X_1|X_2)$, introduce variables $q_{3,3}(X_1, X_2)$, related to the p_k by multilinear constraints in Figure 2.c. These new variables are subject to constraints in Figure 2.d. We must also introduce variables $q_{3,\hat{3}}(X_1, X_2)$, subject to constraints that are identical to those in Figures 2.c and 2.d. — except that 3 is everywhere replaced by $\hat{3}$. □

The previous example can be easily extended to binary Markov chains with n nodes.[3] The number of multilinear constraints generated by the procedure at random variable X_i, which we denote by $T(i)$, is recursively expressed as $T(i) = \mathcal{O}(2^i) + 2T(i-1)$, thus we have $T(i) = \mathcal{O}(i2^i)$ (the number of linear constraints follows a similar pattern). The total number of multilinear constraints is of order $\sum_{i=1}^{n} \mathcal{O}(i2^i)$, and thus of order $\mathcal{O}(n2^n)$. Given the inherent complexity of epistemic independence, this exponential growth is not surprising in exact calculations. However we can be more positive about the MULTILINEAREXTENSION algorithm.

First, even if the algorithm cannot deal with large networks, it does allow us to address non-trivial networks — certainly larger networks than the ones handled by the E^3 algorithm. Consider a Markov chain with 5 nodes, X_1 to X_5. The MULTILINEAREXTENSION algorithm leads to 152 multilinear constraints, a number that can be easily handled by existing multilinear programming algorithms [13]. On the other hand the E^3 algorithm cannot go beyond a Markov chain with 4 nodes — because the algorithm requires explicit manipulation of epistemic extensions, and the extension of a Markov chain with 4 binary nodes typically contains millions of extreme points.

Second, the MULTILINEAREXTENSION algorithm generates a program with a rather modular structure that can be explored by approximation techniques. Standard approximations from multilinear programming can be used [20, 29], or approximations that are specific to epistemic extensions can be investigated. The E^3 algorithm offers no such path.

Third, depending on the independence relations expressed in a network, several simplifications may be possible — as illustrated by the next example.

Example 4 Consider the network in Figure 3, taken from [7]. To process X_1, we must enforce $EIN(X_1, (X_2, X_3, X_4))$: we need 16 constraints and we must then enforce $EIN(X_2, X_3)$ — however this second judgement can be directly enforced without any multilinear constraint. Likewise, we can enforce $EIN(X_1, X_3)$ without any multilinear constraint. When we process X_4, we must enforce $EIN(X_4, (X_1, X_5)|X_2, X_3)$; to do so, we need 32 multilinear constraints and then we must en-

[3] A Markov chain with n nodes has root node X_1 and terminal node X_n, such that every node X_i between them has a single parent X_{i-1} and a single child X_{i+1}; X_1 has a single child and X_n has a single parent.

force (among other things) $EIN(X_3, (X_1, X_2, X_5))$ and $EIN(X_3, X_5|X_1, X_2)$ — however the latter judgement is redundant as it is implied by the former. □

Our experience indicates that multilinear programs with a few thousand variables can be solved with existing hardware, thus indicating that a (not too dense) network containing about 10 to 12 nodes can be processed in reasonable time. The limits of the algorithm depend on the network topology (the density of connections in the network) but also on the number of values of variables and the complexity of the local credal sets. Even though the viable networks are still small, they can serve as testing ground for approximate algorithms to be developed in the future.

Finally, consider the following question: given a joint probability $P(X_1, \ldots, X_n)$, does this measure belong to the epistemic extension of a network or not? With the E[3] algorithm, the only way to answer this question is to construct the whole extension and then test for inclusion. The multilinear formulation offers a more viable route, as we can test whether a sequence of multilinear programs are satisfied or not. The existence of an "inclusion test" may lead to algorithms that generate distributions and test for inclusion, detecting possible problems and modifying distributions gradually — we leave this path for the future. We close this section by noting that the algorithm is "incremental" in the sense that constraints are built in blocks, and a new irrelevance judgement can be added with relatively "local" changes on existing constraints already built by the algorithm[4]

5 Separation properties

In a Bayesian network, the computation of a conditional probability $P(Q|E)$ typically does not require manipulation of all nodes in the network [16]. Call *evidence* the set of random variables X_i that have their values fixed by the event E. There are two kinds of nodes that can be discarded given Q and E: *barren nodes* and "top" nodes that are separated from Q by the evidence in the *moral graph* [27].[5] In a Bayesian network, the value of $P(Q|E)$ can be obtained in the sub-network without barren and separated nodes and without nodes that define E. These separation properties have been elegantly condensed into the criterion of *d-separation*, an algorithmically simple (polynomial) test that detects independence in Bayesian networks [25]. However, the proof of soundness of d-separation depends on the *semi-graphoid* properties of stochastic independence [12, 16, 25, 28]. The problem here is that one

[4]We thank a reviewer for bringing this property of the algorithm to our attention.

[5]A node X_i is a *barren node* if it is not used to define events Q and E, and either it has no descendants, or its descendants are also barren nodes. The moral graph of a Bayesian network is obtained by connecting all parents of nodes and then removing the direction of all edges.

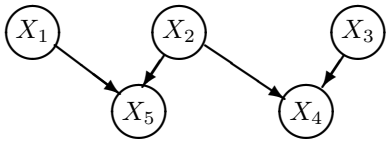


Figure 3: Example 4.

of the semi-graphoid properties, the *contraction* property, fails for epistemic independence [11].

Can separation properties of Bayesian networks be extended to epistemic extensions based on irrelevance/independence? Some results are known: barren nodes can be removed from a credal network to compute epistemic extensions based on irrelevance/independence [7]. In the next theorem we focus on separation in Markov chains — the theorem shows that evidence in a node X_j makes "upstream" nodes independent of "downstream" nodes.

Theorem 2 *Consider a Markov chain with n nodes, with separately specified local credal sets $K(X_1)$ and $K(X_i|\mathrm{pa}(X_i))$ for $i > 1$, such that no conditioning event has zero lower probability. For $i < j < k$, $EIN(X_i, X_k|X_j)$ in the epistemic extension based on independence.*

Proof. Consider first $EIR(X_k, X_i|X_j)$ and the following inductive argument. If $k = j + 1$, the irrelevance is trivial: the Markov condition leads to $EIR(X_k, (X_1, \ldots, X_{j-1})|X_j)$, and the direct decomposition property (a graphoid property [11]) can be used to remove X_1 to X_{j-1}, except X_i. Now consider $j + l$ for $l > 0$; assume $EIR(X_{j+l}, X_i|X_j)$. The Markov condition and direct decomposition imply $EIR(X_{j+l+1}, (X_i, X_j)|X_{j+l})$; by direct weak union, $EIR(X_{j+l+1}, X_i|(X_{j+l}, X_j))$. By reverse contraction, $EIR(X_{j+l}, X_i|X_j)$ and $EIR(X_{j+l+1}, X_i|(X_{j+l}, X_j))$ imply $EIR((X_{j+l}, X_{j+l+1}), X_i|X_j)$, and X_{j+l} can be removed by reverse decomposition. The result is obtained when $j + l + 1 = k$.

Now consider $EIR(X_i, X_k|X_j)$. Again the result is trivial for $k = j + 1$. We follow the same inductive argument: we assume $EIR(X_i, X_{j+l}|X_j)$ and we want $EIR(X_i, (X_{j+l}, X_{j+l+1})|X_j)$. However we cannot use contraction here [11], so we must take a different route. Take an arbitrary function $f(X_{j+l}, X_{j+l+1})$; to simplify notation, we use r for $j + l$. By selecting the following distribution, clearly independent of X_i and X_j:

$$P(X_{r+1}|X_1, \ldots, X_r) = \arg\min E[f(X_{r+1}, X_r)|X_r]\,, \tag{6}$$

we have:

$$\begin{aligned}
&\underline{E}[f(X_r, X_{r+1})|X_i, X_j] \\
&\quad = \min E[E[f(X_r, X_{r+1})|X_i, X_j, X_r]\,|X_i, X_j] \\
&\quad = \min E[\underline{E}[f(X_r, X_{r+1})|X_r]\,|X_i, X_j] \\
&\quad = \underline{E}[\underline{E}[f(X_r, X_{r+1})|X_r]\,|X_i, X_j]\,.
\end{aligned}$$

By assumption $EIR(X_i, X_r|X_j)$, thus the previous iterated lower expectation is equal to $\underline{E}[\underline{E}[f(X_r, X_{r+1})|X_r]\,|X_j]$, and then $\underline{E}[f(X_r, X_{r+1})|X_i, X_j] = \underline{E}[\underline{E}[f(X_r, X_{r+1})|X_r]\,|X_j]$. Note that $\underline{E}[f(X_r, X_{r+1})|X_i, X_j]$ cannot be smaller than this last iterated lower expectation [33]. Likewise,

$$\begin{aligned}
\underline{E}[f(X_r, X_{r+1})|X_j] &= \min E[\underline{E}[f(X_r, X_{r+1})|X_r]\,|X_j] \\
&= \underline{E}[\underline{E}[f(X_r, X_{r+1})|X_r]\,|X_j] \\
&= \underline{E}[f(X_r, X_{r+1})|X_i, X_j]\,.
\end{aligned}$$

This argument requires that minimizing distributions be actually available in the epistemic extension. To see that this is the case, consider the auxiliary extension generated by multiplying every distribution in the epistemic extension $K(X_1, \ldots, X_r)$ by the distribution in Expression (6). The resulting extension does satisfies the Markov condition for $X_1, \ldots, X_r$ and also for X_{r+1} (because Expression (6) defines the conditional of X_{r+1} given $X_1, \ldots, X_r$, and this distribution is independent of $X_1, \ldots, X_{r-1}$). Thus the auxiliary extension belongs to the epistemic extension, and it contains an appropriate minimizing probability distribution. As $f(X_r, X_{r+1})$ is arbitrary, we obtain $EIR(X_i, (X_{j+l+1}, X_{j+l})|X_j)$ and then $EIR(X_i, X_{j+l+1}|X_j)$ by direct decomposition. □

The theorem only considered extensions based on epistemic independence, and focused on a relatively simple independence relation on chains. It is possible that the proof can be extended to much more complex networks and more general relations without much change; however it seems that a substantially new approach would be needed to prove full d-separation in case it is valid in the present context. It is possible that, even though full d-separation is not valid, some simpler (possibly asymmetric) separation property is valid.[6]

6 Conclusion

Epistemic irrelevance and independence arguably offer the "right" way to define a behavioral notion of independence for credal sets. However, these concepts are difficult to manipulate computationally. On the one hand, judgements of epistemic irrelevance and independence lead to very complex joint credal sets; on the other hand, little is known about their separation properties and other simplifications that are routinely applied with stochastic independence. In this paper we have contributed with techniques and results that increase the current understanding about epistemic irrelevance and independence.

First, we have presented multilinear programming methods that handle general judgements about events, and judgements about random variables expressed through credal networks. These techniques open the possibility that approximation methods from multilinear programming can be profitably adapted, something that cannot be easily done with existing methods. Also, algorithms inherit convergence guarantees from multilinear programming — it is an open question whether such guarantees can be given for Walley's algorithm and its generalizations. Moreover, our algorithms are more efficient than existing methods, particularly for manipulation of random variables, because they do not require explicit construction of extensions. The multilinear formulation even opens the possibility of mixing judgements of epistemic and strong independence in the same algorithmic framework. Certainly we leave many avenues for further work; for example, a precise characterization of computational complexity for epistemic irrelevance and independence is still open.

Second, we have shown that usual separation properties employed in Bayesian networks hold for Markov chains. Many important properties of stochastic independence have no known analogues for epistemic irrelevance and independence; an interesting avenue for further is exactly to find such analogues.

Acknowledgements

We thank Peter Walley for sharing with us the algorithm in Figure 4. This work has received generous support from HP Brazil R&D. The work has also been supported by CNPq (through grant 3000183/98-4) and FAPESP.

A Walley's iterative algorithm

The iterative procedure described in Figure 4 produces inferences for an event D conditional on another event E, under judgements of epistemic irrelevance. The method has been conceived by Walley (personal communication); we present a very brief summary so as to compare it to our multilinear programming approach. The idea of Walley's algorithm is to check, at each iteration, whether irrelevance assessments are satisfied by a pool of constraints; if not, then the smallest change in assessments that can lead to satisfaction of irrelevance judgements is computed and the current constraints are modified accordingly. Each iteration modifies at least one of current assessments (or stops). The algorithm gradually converges to a set of con-

[6] We thank a reviewer for bringing this possibility to our attention, as well as for pointing out the relevant references [23] and [30].

straints that represent the whole natural extension. Obvious changes to Walley's algorithm can account for assessments containing random variables and for judgements of conditional irrelevance among events.

It is also possible to conceive an extension of the algorithm so as to handle judgements of irrelevance among random variables, even though its practical feasibility is unclear at the moment. So as to facilitate comparison with our methods, we outline one such extension here. Consider judgements of the form $EIR(X_j, Y_j|C)$. We start by collecting all assessments (other than judgements of independence). For each judgement $EIR(X_j, Y_j|C)$, we obtain an explicit description of $K(Y_j|C)$ and of $K(Y_j|X_j = x, C)$; this has the same purpose of step (2.2) in Walley's algorithm. To generate an explicit description of $K(Y_j|C)$ or $K(Y_j|X_j = x, C)$, we must resort either to Fourier-Motzkin elimination or to an enumeration procedure [19]. If $K(Y_j|C)$ and $K(Y_j|X_j = x, C)$ have the same convex hull for every value of X_j, for every j, then we stop (as in the "first half" of step (2.3)). Suppose that, for a given j, $K(Y_j|C)$ and $K(Y_j|X_j = x, C)$ have different convex hulls. Now we simply enforce that each one of these sets must satisfy all constraints in their current intersection (take the union of constraints defining these sets) — this is similar to the "second half" of step (2.3) in Walley's algorithm. The procedure just outlined gradually constructs the whole natural extension. A computer implementation would have to struggle with several difficulties: first, the explicit description of sets $K(Y_j|C)$ and $K(Y_j|X_j = x, C)$ may require an exponential growth in the number of constraints; second, it is not easy to detect when sets have identical convex hulls; finally, it is not clear that this extended algorithm is always convergent, let alone finitely convergent.

References

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

[2] 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.

[3] A. Capotorti, G. Coletti, R. Scozzafava. Algorithms for processing partial probabilistic assessments. *Third Congres Europeene des Systemes*, pages 506–511, Kappa, 1996.

[4] G. Coletti and R. Scozzafava. Stochastic independence in a coherent setting. *Annals of Mathematics and Artificial Intelligence*, 35:151–176, 2002.

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

[6] F. G. Cozman. Calculation of posterior bounds given convex sets of prior probability measures and likelihood functions. *Journal of Computational and Graphical Statistics*, 8(4):824–838, 1999.

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

[8] F. G. Cozman. Separation properties of sets of probabilities. In *Proc. of Conf. on Uncertainty in Artificial Intelligence*, pages 107–115, San Francisco, 2000. Morgan Kaufmann.

[9] F. G. Cozman. Algorithms for conditioning on events of zero lower probability. In *Proc. of the Fifteenth Int. Florida Artificial Intelligence Research Society Conf.*, pages 248–252, Pensacola, Florida, 2002.

[10] F. G. Cozman. Computing lower expectations with Kuznetsov's independence condition. In *Proc. of the Third Int. Symp. on Imprecise Probabilities and Their Applications*, pages 177–187, Lugano, Switzerland, 2003. Carleton Scientific.

[11] F. G. Cozman and P. Walley. Graphoid properties of epistemic irrelevance and independence. In *Proc. of the Second Int. Symp. on Imprecise Probabilities and Their Applications*, pages 112–121, Ithaca, New York, 2001.

[12] A. P. Dawid. Conditional independence. In *Encyclopedia of Statistical Sciences, Update Volume 2*, pages 146–153. Wiley, New York, 1999.

[13] C. P. de Campos and F. G. Cozman. Inference in credal networks using multilinear programming. In *Proc. of the Second Starting AI Researchers' Symp. (STAIRS)*, pages 50–61, Amsterdam, The Netherlands, 2004. IOS Press.

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

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

[16] D. Geiger, T. Verma, and J. Pearl. Identifying independence in Bayesian networks. *Networks*, 20:507–534, 1990.

[17] F. J. Giron and S. Rios. Quasi-Bayesian behaviour: A more realistic approach to decision making? In *Bayesian Statistics*, pages 17–38. University Press, Valencia, Spain, 1980.

[18] W. Gochet and Y. Smeers. A branch-and-bound method for reversed geometric programming. *Operations Research*, 27(5):983–996, 1979.

(1) Set $\nu_j = 0$ and $\mu_j = 1$ for all j.
(2) Repeat:
(2.1) Form the constraints $\mathcal{C}_0$ as in Section 3, add the unitary constraint $\sum_k p_k = 1$.
(2.2) With the constraints in the previous step, update (using fractional linear programming):
$\nu'_j = \underline{P}(B_j|A_j, C_j)$, $\nu''_j = \underline{P}(B_j|A_j^{\complement}, C_j)$, $\nu^*_j = \underline{P}(B_j|C_j)$, and
$\mu'_j = \overline{P}(B_j|A_j, C_j)$, $\mu''_j = \overline{P}(B_j|A_j^{\complement}, C_j)$, $\mu^*_j = \overline{P}(B_j|C_j)$ for all j.
(2.3) If $(\nu'_j = \nu''_j = \nu^*_j)$ and $(\mu'_j = \mu''_j = \mu^*_j)$ for all j, stop;
Otherwise, take $\nu_j = \max(\nu'_j, \nu''_j, \nu^*_j)$ and $\mu_j = \min(\mu'_j, \mu''_j, \mu^*_j)$ for all j, and return to (2.1).
(3) Using the assessments reached when step (2.3) breaks the loop, compute $\max P(D|E)$ using linear fractional programming.

Figure 4: Walley's method for inferences with epistemic irrelevance among events.

[19] P. Hansen and B. Jaumard. Probabilistic satisfiability. Technical Report G-96-31, Les Cahiers du GERAD, École Polytechique de Montréal, 1996.

[20] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer-Verlag, 1995.

[21] V. P. Kuznetsov. *Interval Statistical Methods*. Radio i Svyaz Publ., (in Russian), 1991.

[22] C.D. Maranas and C.A. Floudas. Global optimization in generalized geometric programming. *Computers and Chemical Engineering*, 21(4):351–370, 1997.

[23] S. Moral. Epistemic irrelevance on sets of desirable gambles. In *Second Int. Symp. on Imprecise Probabilities and Their Applications*, pages 247–254, 2001.

[24] S. Moral and A. Cano. Strong conditional independence for credal sets. *Annals of Mathematics and Artificial Intelligence*, 35:295–321, 2002.

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

[26] H.D. Sherali and W.P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Kluwer Academic Publishers, 1999.

[27] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search (2nd ed.)*. MIT Press, 2000.

[28] M. Studeny. Semigraphoids and structures of probabilistic conditional independence. *Annals of Mathematics and Artificial Intelligence*, 21(1):71–98, 1997.

[29] H. Tuy. *Convex Analysis and Global Optimization*, volume 22 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, 1998.

[30] B. Vantaggi. Graphical models for conditional independence structures In *Second Int. Symp. on Imprecise Probabilities and Their Applications*, pages 332–341, 2001.

[31] B. Vantaggi. Graphical representation of asymmetric graphoid structures. In *Third Int. Symp. on Imprecise Probabilities and Their Applications*, pages 560–574. Carleton Scientific, 2003.

[32] P. Vicig. Epistemic independence for imprecise probabilities. *Int. Journal of Approximate Reasoning*, 24:235–250, 2000.

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