Geometric Foundations for Interval-Based Probabilities

Vu Ha [a] AnHai Doan [b] Van Vu [c] Peter Haddawy [a]

[a] *Decision Systems and Artificial Intelligence Lab,*
Department of Electrical Engineering & Computer Science,
University of Wisconsin-Milwaukee
Milwaukee, WI 53201
E-mail: {vu,haddawy}@cs.uwm.edu
[b] *Department of Computer Science and Engineering,*
University of Washington
Seatle, WA 98195
E-mail: anhai@cs.washington.edu
[c] *Department of Mathematics,*
Yale University
New Haven, CT 06520
E-mail: vuha@math.yale.edu

The need to reason with imprecise probabilities arises in a wealth of situations ranging from pooling of knowledge from multiple experts to abstraction-based probabilistic planning. Researchers have typically represented imprecise probabilities using intervals and have developed a wide array of different techniques to suit their particular requirements. In this paper we provide an analysis of some of the central issues in representing and reasoning with interval probabilities. At the focus of our analysis is the probability cross-product operator and its interval generalization, the *cc-operator*. We perform an extensive study of these operators relative to manipulation of sets of probability distributtions. This study provides insight into the sources of the strengths and weaknesses of various approaches to handling probability intervals. We demonstrate the application of our results to the problems of inference in interval Bayesian networks and projection and evaluation of abstract probabilistic plans.
Keywords: Imprecise probabilities, convex sets of distributions, abstraction-based probabilitic planning, interval Bayesian networks.

1. INTRODUCTION

In recent years the use of imprecise probabilities has found application in a wealth of areas. Imprecise probabilities can be used to facilitate elicitation when the available domain knowledge is insufficient to specify exact probabilities [10,21], or when eliciting knowledge from multiple experts [19]. They may result from the abstraction of more detailed probabilistic models [2,12], and are useful in studying sensitivity and robustness in probabilistic inference, e.g. in Bayesian networks [4]. The use of imprecise probabilities is even advocated as an alternative representation of belief by researchers who do not feel comfortable with the paradigm of *Strict Bayesianism* which requires exact probabilistic models [18,17,27].

While several methods exist to represent imprecise probabilities, representation using intervals is the most commonly used approach. Researchers working on the problems mentioned in the previous paragraph have developed a number of techniques for handling probability intervals. But to date we lack a uniform and comprehensive study of the central issues concerning representation and manipulation of probability intervals. In this paper, we make the first steps towards such a study. Our approach originates from the observation that the probability cross-product operator lies in the hearts of numerous computations such as probabilistic plan projection, expected utility computation, and Bayesian network propagation. We present an interval generalization of this operator, called the *cc-operator* [1], and provide a throrough analysis of some key properties of the cc-operator relative to manipulation of sets of probability distributions. We then show how the cc-operator can be substituted for the probability cross-product to produce interval versions of plan projection and Bayesian network propagation algorithms. Our Bayesian network propagation algorithm is based on Dechter's Bucket Elimination algorithm [6]. We draw upon our theoretical analysis of the cc-operator to produce efficient versions of these generalized algorithms. Our approach rests on the well-developed area of finite convex geometry. Thus our results are applicable to problems with discrete finite probability distributions.

The rest of this paper is organized as follows. We start Section 2 with a quick review of several convex geometric concepts most relevant to our analysis. We then define and analyze the cc-operator. In Section 3 we define the concept

[1] "cc" stands for "convex combination". It was originally called *affine-operator* in [12]. We adopt this new term in accordance with standard convex geometry terminology.

of cc-trees, which are data structures that have cc-operators as basic building blocks. These data structures, in particular their interval and probability versions, icc-trees and pcc-trees, will be used throughout our analysis. In Section 4, we present the theoretical foundations of an abstraction-based probabilistic planner, DRIPS [13], where the cc-operator plays a fundamental role. In Section 5, we use the concept of icc-trees to develope an interval version of Dechter's Bucket Elimination algorithm for propagation in interval Bayesian networks. For both plan projection and Bayesian network propogation, the probability bounds computed by our algorithms are correct but not tight. We suggest an explanation for the difficulty of computing tight probabilistic bounds in the above problems. In Section 6, we return to analyzing the cc-operator. In particular, we investigate pcc-trees as a new approach to represent convex sets of probability distributions. We show that the class of pcc-trees is identical to the class of polytope-like sets of probability distributions, and that in a special case, Dempster-Shafer belief functions, when viewed as lower envelopes of sets of probability distributions, can be represented by 2-level pcc-trees in at least two different ways. In Section 7, we address the issue of evidential updating with pcc-trees. The in-depth study of the cc-operator provides an in-sightful explanation of why updating with pcc-trees in general and with belief functions in particular is difficult. The latter issue has been a subject of extensive study, and our analysis adds a new perpective to it.

2. PROBABILITY CROSS-PRODUCT AND THE CC-OPERATOR

We start with a brief introduction of some basic concepts of convex geometry. For more details, see [11].

The *d-dimension Euclidean Space* is the d-dimension vector space $\Re^d$ equipped with the inner product $\langle\rangle$, which is defined for any pair of points $x, y \in \Re^d$ as: $\langle x, y\rangle = \sum_{i=1}^{d} x_i y_i$. A *convex combination* of the points $x_1, \ldots, x_n \in \Re^d$ is a linear combination $\lambda_1 x_1 + \cdots + \lambda_n x_n$, where $\lambda_i \geq 0, i = 1, \ldots, n$, and $\lambda_1 + \cdots + \lambda_n = 1$. The coefficients λ_i are called *convex coefficients*, and the sum is denoted by $\overset{cc}{\sum}$. This sum is also called the *probability cross-product* of 2 vectors whose components are λ_i and x_i. For any set $K \subseteq \Re^d$, the *convex hull* of K, denoted by $\mathrm{conv}(K)$, is the set of all convex combinations of the points from K. A set $K \subseteq \Re^d$ is said to be *convex* if $K = \mathrm{conv}(K)$. A mapping ϕ from a convex set $K \subseteq \Re^d$ to $\Re^e$ is said to be a *convex mapping* if it preserves convex combinations, i.e. $\phi(\overset{cc}{\sum}{}_{i=1}^{n}\lambda_i x_i) = \overset{cc}{\sum}{}_{i=1}^{n}\lambda_i\phi(x_i)$, for all points $x_i \in K$ and convex coefficients λ_i. A convex mapping always maps convex sets into convex sets. The convex hull of a finite set of points is called *a polytope*. Given a polytope K, the smallest set of points whose convex hull is equal K is called the set of *vertices* of the polytope. The polytope whose vertices are the points $(1,0,\ldots,0),(0,1,\ldots,0),\ldots,(0,0,\ldots,1)$ is exactly the set $\mathcal{S}$ of all probability distributions over a sample space Ω of cardinality d and is called the *probability simplex.* For example, the probability distribution P over the sample space $\Omega = \{s_1, s_2, \ldots, s_d\}$ corresponds to the point $(P(s_1), P(s_2), \ldots, P(s_d))$ in $\mathcal{S}$.

We now define the generalized version of the probability cross-product, the cc-operator. In the rest of the paper, an interval is implicitly understood as a closed subinterval of $[0,1]$.

Definition 2.1. The *cc-operator* $\otimes$ defined by an interval vector $\vec{\Lambda} = (\Lambda_1, \ldots, \Lambda_n)$ is the function that maps a vector $\vec{w} = (w_1, \ldots, w_n)$ of sets of points in $\Re^d$ to the set $\vec{\Lambda} \otimes \vec{w}$ that consists of all points of the form $\vec{\lambda} \otimes \vec{x} = \overset{cc}{\sum}{}_{i=1}^{n}\lambda_i x_i$, where $\lambda_i \in \Lambda_i, x_i \in w_i, i = 1, \ldots, n$. We sometimes write $\vec{\Lambda} \otimes \vec{w}$ as $\overset{cc}{\sum}{}_{i=1}^{n}\Lambda_i w_i$. In the case when w_i are also intervals, we call this sum the *icc-operator* sum (letter "i" stands for "interval") and denote it by $\overset{icc}{\sum}$.

The cc-operator sum $\vec{\Lambda} \otimes \vec{w}$ is a special case of the Minkowski sum, a familiar notion in convex geometry. The difference between the two sums is that the coefficients λ_i in the Minkowski sum are arbitrary. Below are a few properties of the cc-operator, which constitute the basis of our framework.

Fact 1. Any probability distribution $P \in \mathcal{S}$ can be expressed as a cc-operator sum as $P = (P(s_1), \ldots, P(s_d)) \otimes (s_1, \ldots, s_d)$, where s_i represents the ith vertex of the probability simplex.

Fact 2. If w_i are sets of probability distributions, then $\vec{\Lambda} \otimes \vec{w}$ is also a set of probability distributions. In other words, $\otimes$ *is closed on* $2^{\mathcal{S}}$.

Fact 3. If each of the intervals Λ_i is the entire interval $[0,1]$, we denote $\vec{\Lambda} \otimes \vec{w}$ as $cc(\vec{w})$, or $cc(w_1, \ldots, w_n)$. If K is a polytope with vertices $\{x_1, \ldots, x_n\}$, then $K = \mathrm{conv}(\{x_1, \ldots, x_n\}) = cc(\{x_1\}, \ldots, \{x_n\})$.

Fact 4. *The cc-operator and convex mappings commute.* Specifically, suppose that $K \subseteq \Re^d$ is a convex set and $\phi : K \to \Re^e$ is a convex mapping. For any $H \subseteq K$, defined $\phi(H)$ as $\phi(H) = \{\phi(x) | x \in H\}$. Then

$$\phi\Big(\overset{cc}{\sum}{}_{i=1}^{n} \Lambda_i w_i \Big) = \overset{cc}{\sum}{}_{i=1}^{n} \Lambda_i \phi(w_i).$$

Theorem 2.2. *The cc-operator and convex hull operator commute.* Specifically:

$$\operatorname{conv}\left(\overset{cc}{\sum}{}_{i=1}^{n} \Lambda_i w_i \right) = \overset{cc}{\sum}{}_{i=1}^{n} \Lambda_i \operatorname{conv}(w_i).$$

Proof. Let $K = \overset{cc}{\sum}{}_{i=1}^{n} \Lambda_i w_i$. To prove the "$\subseteq$" relation, let us consider a convex combination of elements of K: $y = \overset{cc}{\sum}{}_{j=1}^{k} \eta_j y_j$ $(y_j \in K)$. Each of the points $y_j \in K$ can be written as $y_j = \overset{cc}{\sum}{}_{i=1}^{n} \lambda_{ji} x_{ji}$, where $\lambda_{ji} \in \Lambda_i$ and $x_{ji} \in w_i$. Then

$$\begin{aligned} y = \overset{cc}{\sum}{}_{j=1}^{k} \eta_j y_j &= \overset{cc}{\sum}{}_{j=1}^{k} \eta_j \left(\overset{cc}{\sum}{}_{i=1}^{n} \lambda_{ji} x_{ji} \right) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} \eta_j \lambda_{ji} x_{ji} \\ &= \overset{cc}{\sum}{}_{i=1}^{n} \left(\overset{cc}{\sum}{}_{j=1}^{k} \eta_j \lambda_{ji} \right) \overset{cc}{\sum}{}_{j=1}^{k} \frac{\eta_j \lambda_{ji}}{\sum_{j=1}^{k} \eta_j \lambda_{ji}} x_{ji} \\ &\in \overset{cc}{\sum}{}_{i=1}^{n} \Lambda_i \operatorname{conv}(w_i). \end{aligned}$$

To prove the "$\supseteq$" relation, consider an element $y = \overset{cc}{\sum}{}_{i=1}^{n} \lambda_i y_i$ of the right hand side, where $\lambda_i \in \Lambda_i, y_i \in \operatorname{conv}(w_i)$. We need to show that $y \in \operatorname{conv}(K)$.

Observe that since $y_1 \in \operatorname{conv}(w_1)$, y_1 can be expressed as $y_1 = \overset{cc}{\sum}{}_{j_1=1}^{l} \eta_{1j_1} x_{1j_1}$ where $x_{1j_1} \in w_1, \forall j_1 = 1, \ldots, l$. Thus

$$\begin{aligned} y = \overset{cc}{\sum}{}_{i=1}^{n} \lambda_i y_i &= \lambda_1 \left(\overset{cc}{\sum}{}_{j_1=1}^{l} \eta_{1j_1} x_{1j_1} \right) + \sum_{i=2}^{n} \lambda_i y_i \\ &= \overset{cc}{\sum}{}_{j_1=1}^{l} \eta_{1j_1} \left(\lambda_1 x_{1j_1} + \sum_{i=2}^{n} \lambda_i y_i \right) \end{aligned}$$

From the above equality, we see that in order to show that $y \in \operatorname{conv}(K)$, it is enough to show that $\lambda_1 x_{1j_1} + \sum_{i=2}^{n} \lambda_i y_i \in \operatorname{conv}(K)$. Repeating the above lines of reasoning $(n-1)$ times, the problem is eventually reduced to showing that $\lambda_1 x_{1j_1} + \lambda_2 x_{2j_2} + \cdots + \lambda_n x_{nj_n} \in \mathrm{conv}(K)$, which is trivial since the left side is actually an element of K and thus must be in $\mathrm{conv}(K)$. □

3. CC-TREES, ICC-TREES, PCC-TREES

In this section we introduce the concept of cc-trees and its special cases: icc-trees and pcc-trees. These trees can be viewed as data structures that encode, among other things, sets of real numbers and sets of probability distributions.

A *cc-tree* is a rooted, annotated tree where the branches are annotated with intervals and the nodes are annotated with sets of points in $\Re^d$, with the constraint that for any internal (i.e. non-leaf) node N with children $\{C_1, \ldots, C_k\}$, the set $\mathcal{D}(N)$ that annotates N is the result of applying the cc-operator defined by the intervals Λ_i that annotate the branches (N, C_i) to the sets $\mathcal{D}(C_i)$ that annotate the children C_i: $\mathcal{D}(N) = \overset{cc}{\sum}{}_{i=1}^{k} \Lambda_i \mathcal{D}(C_i)$. Thus, a cc-tree is completely defined by specifying the sets of points associated with the leaves and the intervals associated with the branches. Intuitively, this tree encodes the set of points obtained by applying the cc-operator from the leaves up toward the root.

Cc-trees whose leaves are annotated with intervals are called *icc-trees*. Figure 1 depicts an example of an icc-tree. We now discuss the evaluation of the icc-operator sum $\overset{icc}{\sum}{}_{i=1}^{n} \Lambda_i w_i$, where $\Lambda_i \subseteq [0\ 1]$ and $w_i \subseteq \Re$ are closed intervals. Note that the set of convex sets of real numbers is exactly the set of intervals. It then follows directly from Theorem 2.2 (cc-operator and convex hull operator commute) that $\overset{icc}{\sum}{}_{i=1}^{n} \Lambda_i w_i$ is also an interval. We show that it is a closed interval by showing how the lower and upper bounds are explicitly obtained. Due to symmetry, we will consider only the upper bound.

Obviously, the upper bound must be achieved by maximizing $\overset{cc}{\sum}{}_{i=1}^{n} \lambda_i u_i$, where $\lambda_i \in \Lambda_i$ and u_i are upper bounds of w_i. This is a special case of the continuous knapsack problem: to pack materials from n different categories into a unit size knapsack, where material from category number i has value u_i and weight restriction Λ_i (meaning that the weight must be in the interval Λ_i). The objective is to maximize the total value, which can be achieved using a greedy approach as follows. We first sort the values u_i in *descending* order, breaking ties arbitrarily. Then proceeding in this order, we try to put into the knapsack as much material of the current category as possible, subjected to two constraints:

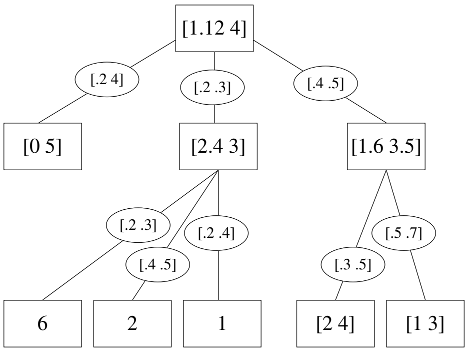

Figure 1. Example of an icc-tree.

1) the weight must be in the constraint interval Λ_i and 2) The sum of the already assigned weights must be at most one minus the sum of the lower bounds of the remaining constraint intervals [2]. The first restriction is explicitly mandatory from the description of the problem, while the second one ensures that the lower bound condition will be satisfied in the future. We illustrate this idea through an example.

Example 3.1. Let us compute $\overset{icc}{\sum}{}_{i=1}^{3}\Lambda_i w_i$, where $\Lambda_1 = [.2\ .4]$, $\Lambda_2 = [.2\ .3]$, $\Lambda_3 = [.4\ .5]$, $w_1 = [0\ 5]$, $w_2 = [2.4\ 3]$, $w_1 = [1.6\ 3.5]$. Thus $u_1 = 5$, $u_2 = 3$, $u_3 = 3.5$. First, we assign weight to material number 1, which is computed as $\min\{.4, 1 - .2 - .4\} = .4$. The remaining weight is thus $1 - .4 = .6$. We then assign weight to material number 3: $\min\{.5, .6 - .2\} = .4$. And finally, the weight assigned to the last material is $1 - .4 - .4 = .2$, and thus total value is $.4\times 5 + .4\times 3.5 + .2\times 3 = 4$. The lower bound is 1.12. Thus $\overset{icc}{\sum}{}_{i=1}^{3}\Lambda_i w_i = [1.12\ 4]$. This icc-sum is represented in Figure 1 by the tree of depth 1 that contains the root and its 3 children.

Remark 3.2. The complexity of the above algorithm to compute icc-operator sum is dominated by the sorting of the material values, which is $O(n \log n)$.

Of our particular interest are cc-trees whose leaves are annotated with the

[2] Tessem [25] called this step *reinforcement*, and the analogous step in computing the lower bound *annihilation*.

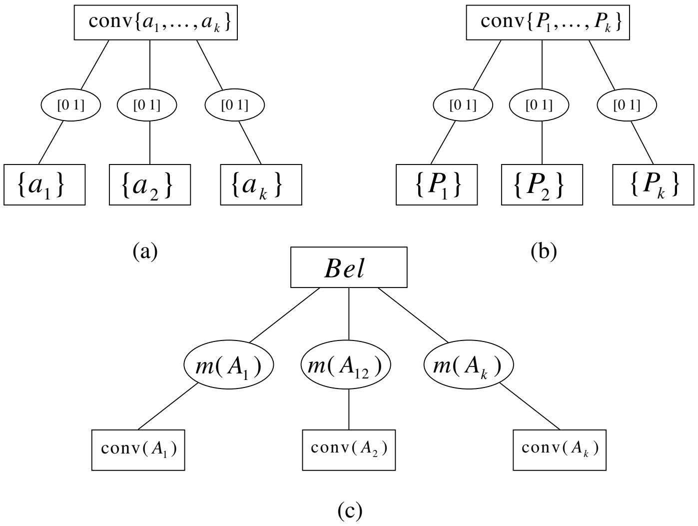

Figure 2. Examples of pcc-trees: (a) The face $\mathrm{conv}(A)$ of the probability simplex where $A = \{a_1, \ldots, a_k\} \subseteq \Omega$; (b) A polytope-like set of distributions with vertices $\{P_1, \ldots, P_k\}$; (c) A Dempster-Shafer belief function Bel with mass function m and focal elements $\{A_1, \ldots, A_k\}$ represented as a 2-level pcc-tree.

vertices of the probability simplex (or more precisely, sets which contain a single element that is a vertex of the probability simplex). Such trees encode sets of probability distributions and will be called *pcc-trees* (the *"p"* letter stands for "probability").

Figure 2 depicts several pcc-trees. Suppose that $A = \{a_1, \ldots, a_k\} \subseteq \Omega$ is a set of vertices of the probability simplex $\mathcal{S}$. The pcc-tree in Figure 2.a represents $\mathrm{conv}(A)$, which is called a *face* of the polytope $\mathcal{S}$. The pcc-tree in Figure 2.b represents a polytope-like sets of probability distributions, which is a polytope with vertices $P_i \in \mathcal{S}$. The pcc-tree in Figure 2.c represents a Dempster-Shafer belief function viewed as the lower envelope of a set of probability distributions. This pcc-tree will be discussed later in Section 6.

4. ABSTRACTION-BASED PROBABILISTIC PLANNING

In the framework of decision-theoretic planning, uncertainty in the state of the world and in the effects of actions are represented with probabilities, and the planner's goals, as well as tradeoffs among them, are represented with utilities.

Given this representation, the objective is to find an optimal plan or policy, where optimality is defined as maximizing expected utility.

In most of the existing decision-theoretic planning approaches, the world is represented with a probability distribution over the state space, and actions are represented as stochastic mappings among the states [14,5,1,26]. Given this framing of the problem, all probabilistic and decision-theoretic planners face the burden of computational complexity in seeking an optimal or near-optimal solution. One popular way to address this problem is to use abstraction techniques to guide the search through the plan space and to reduce the cost of plan evaluation. This concept has been applied in Markov process-based planning [1] as well as less structured approaches [14].

In the framework of DRIPS, the planning problem is formalized as the problem of searching for the optimal plan through the space of candidate plans, each of which is a sequence of actions. The planner is armed with the capability of understanding and evaluating abstract plans constructed from abstract actions. An abstract action is defined as a mapping from a probability distribution over the state space to a *set* of probability distributions over the state space, and is viewed as a *sound abstraction* of several concrete actions. By evaluating abstract plans, the planner is able to evaluate and eliminate a set of suboptimal plans without explicitly evaluating each member of that set. Herein lies the computational efficiency of DRIPS.

We now present the underlying theoretical framework of DRIPS, where the cc-operator and the derived cc-trees play fundamental roles.

4.1. PROBABILISTIC ACTIONS AND PLANS

A *probabilistic action* A is defined as a function $A : \Omega \rightarrow \mathcal{S}$ that maps a state $s \in \Omega$ of the world to a probability distribution $A(s)$ over possible states of the world. For example, the action in Figure 3.a maps state s_1 to the probability distribution $A(s_1)$ that assigns s_1 the probability of .4 and s_2 the probability of .6, and maps states s_2 and s_3 to some other distributions $A(s_2), A(s_3)$ over $\Omega = \{s_1, s_2, s_3\}$. The distributions $A(s), s \in \Omega$ are called the *branches of action* A. This function A is then extended to a function that maps a probability distribution $P \in \mathcal{S}$ to a probability distribution $A(P) = E_P[A(s)] = \overset{cc}{\sum}_{s\in\Omega} P(s)A(s)$, which is essentially the probability cross-product of the distribution P and the action A.

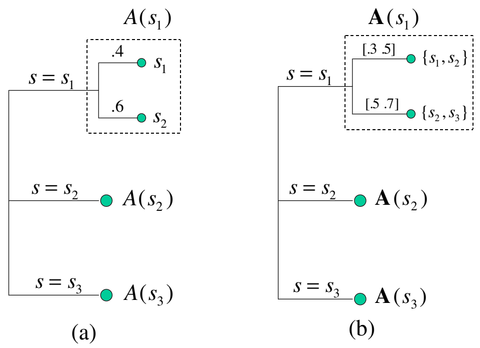

Figure 3. Example of (a) a concrete action and (b) an abstract action.

A *probabilistic plan* pl is a finite sequence of probabilistic actions: $pl = \{A_1, \ldots, A_n\}$. Given an initial world $w_{ini} \in \mathcal{S}$ that is a probability distribution over the state space Ω, the *projection* of pl on w_{ini} is the process of determining the final world $w_{final} = pl(w_{ini}) = A_n(\ldots A_1(w_{ini})\ldots) \in \mathcal{S}$.

A *utility function* is a function $u : \Omega \to \Re$ that assigns to each state of the world a real number that indicates the desirability of that state. The expected utility of a plan pl is defined as $E_u(pl) = \overset{cc}{\sum}_{s\in\Omega} w_{final}(s)u(s)$, where $w_{final} = pl(w_{ini})$. It is convenient to extend the definition of the utility function u to a function $u : \mathcal{S} \to \Re$, where $u(P) = E_P[u(s)] = \overset{cc}{\sum}_{s\in\Omega} P(s)u(s), \forall P \in \mathcal{S}$. Now, the expected utility of a plan pl, executed on an initial world w_{ini} can be written as $u(w_{final}) = u(pl(w_{ini}))$. A plan pl is *optimal* if it has the maximum expected utility among all candidate plans.

Probabilistic actions and utility functions, as functions that have $\mathcal{S}$ as their domains, have the following important property.

Theorem 4.1. Probabilistic actions, probabilistic plans, and utility functions are convex mappings and thus commute with the cc-operator (Fact 4).

Proof. Note that both probabilistic actions and utility functions are first defined as functions over Ω, and then extended by means of expectation to functions over $\mathcal{S}$. Let $\phi : \Re^d \supseteq \mathcal{S} \to \Re^e$ be any function defined and extended in the same way, i.e. $\phi(P) = \overset{cc}{\sum}_{s\in\Omega} P(s)\phi(s), \forall P \in \mathcal{S}$. We show that ϕ is a convex mapping as follows:

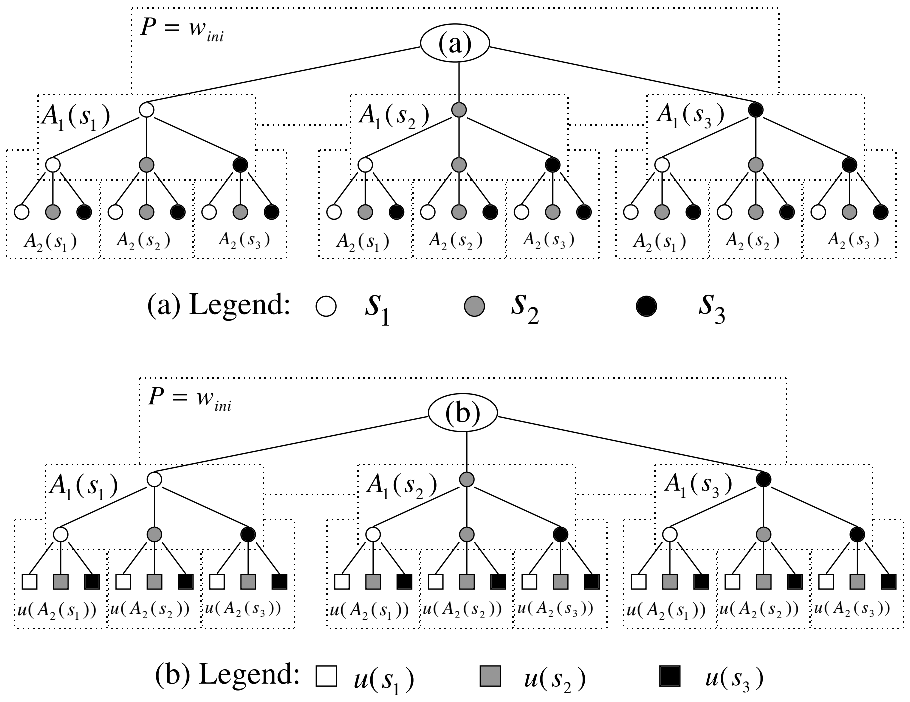

Figure 4. Illustration of (a) projecting and (b) evaluating probabilistic plans.

$$
\begin{aligned}
\phi\Big(\overset{cc}{\sum}{}_{i=1}^{n}\lambda_i P_i\Big) &= \overset{cc}{\sum}{}_{s\in\Omega}\left(\overset{cc}{\sum}{}_{i=1}^{n}\lambda_i P_i\right)(s)\phi(s)\\
&= \overset{cc}{\sum}{}_{s\in\Omega}\left(\overset{cc}{\sum}{}_{i=1}^{n}\lambda_i P_i(s)\right)\phi(s)\\
&= \overset{cc}{\sum}{}_{i=1}^{n}\lambda_i \ \overset{cc}{\sum}{}_{s\in\Omega}P_i(s)\phi(s)\\
&= \overset{cc}{\sum}{}_{i=1}^{n}\lambda_i\phi(P_i).
\end{aligned}
$$

Thus, probabilistic actions and utility functions are convex mappings. It is easy to see that compositions of convex mappings are also convex mappings, and hence probabilistic plans are also convex mappings. □

Figure 4 illustrates the implications of Theorem 4.1 on the process of projecting and evaluating probabilistic plans. Consider projecting a 2-action plan

$pl = \{A_1, A_2\}$ on the initial world $w_{ini} = P$. The state space Ω consists of three elements $\{s_1, s_2, s_3\}$. The initial world is represented by a pcc-tree with three leaves s_i and three branches $P(s_i)$ (Fact 1). In Figure 4.a, this is the pcc-tree with root (a) and nodes of depth 1. Now consider the world after executing action A_1, namely $A_1(P)$. Since A_1 commutes with the cc-operator defined by P, $A_1(P)$ can be represented by the pcc-tree with root (a) and nodes of depth 1 and 2. We in effect obtained this tree ($A_1(P)$) by replacing the leaves s_i of the pcc-tree P by distributions $A_1(s_i)$. Similarly, we can obtain the pcc-tree representing $A_2(A_1(P))$ by replacing the leaves s_i of the pcc-tree $A_1(P)$ with $A_2(s_i)$. This pcc-tree is the whole tree of depth 3 in Figure 4.a. To compute the expected utility of pl, which is $u(A_2(A_1(P)))$, we can replace the leaves s_i of the pcc-tree representing $A_2(A_1(P))$ with $u(s_i)$ (due to the fact that utility function commute with the cc-operator) and then evaluate the obtained icc-tree (Figure 4.b).

4.2. ABSTRACT ACTIONS AND PLANS

From now on, we will call probabilistic actions *concrete actions*. We generalize concrete actions into *abstract actions*, according to [12], by allowing the probabilities to be interval-values instead of point-values, and the resulting worlds to be sets of states instead of states. This definition is best understood through an example (see Figure 3.b). An abstract action is interpreted as a set of concrete actions each of which is obtained by *instantiation*, a process that replaces the probability intervals with consistent probability numbers, and sets of states with consistent states. The consistency condition for probability numbers means that they must be in the corresponding intervals, and obey the axioms of probability. The consistency conditions for states means that they must be in the corresponding sets. For example, the concrete action A in Figure 3.a is an instantiation of the abstract action $\mathbf{A}$ in Figure 3.b. We denote the instantiation relationship as $A \in \mathbf{A}$. The semantics of an abstract action $\mathbf{A}$ is that it is a function $\mathbf{A} : \mathcal{S} \to 2^{\mathcal{S}}$ that maps a probability distribution $P \in \mathcal{S}$ to a set of probability distributions $\{A(P) | A \in \mathbf{A}\}$.

An *abstract plan* **pl** is a sequence of actions, among which some actions may be abstract actions. An *instantiation plan* pl of an abstract plan **pl** is obtained by replacing each abstract action in **pl** with one of its instantiated actions. We denote this relationship as $pl \in \mathbf{pl}$. The projection of an abstract plan **pl** on an initial world w_{ini} is the process of determining the final world w_{final}, which is defined as: $w_{final} = \{pl(w_{ini}) | pl \in \mathbf{pl}\}$. The expected utility of an abstract plan $\mathbf{pl}$ is defined as the $u(\mathbf{pl}(w_{ini})) = \{u(pl(w_{ini})) | pl \in \mathbf{pl}\}$.

These notions of abstract actions and plans are introduced in [12] to model sound abstraction, the central concept of DRIPS. An abstract action $\mathbf{A}$ is called a *sound abstraction* of n concrete actions $A_1, \ldots, A_n$ if for any initial world $w_{ini} \in \mathcal{S}$, we have $A_i(w_{ini}) \in \mathbf{A}(w_{ini}), i = 1, \ldots, n$. For example, the abstract action $\mathbf{A}$ in Figure 3.b is a sound abstraction of the action A in Figure 3.a. The use of abstraction allows DRIPS to evaluate a set of concrete plans, represented as a single abstract plan, in a single step, rather than having to evaluate each of the concrete plans in that set individually. This can substantially reduce the time complexity of finding optimal plans, as demonstrated by a number of toy examples as well as successful real-world applications of DRIPS [13].

Evaluating an abstract plan for this purpose amounts to determining the infimum and the supremum of the set $u(\mathbf{pl}(w_{ini})) \subseteq \Re$, which are exactly the endpoints of the interval $\mathrm{conv}(u(\mathbf{pl}(w_{ini})))$. We call this the *expected utility interval* of the abstract plan $\mathbf{pl}$. This interval can be approximately computed with the help of the next theorem, which is a generalization of Theorem 4.1.

Theorem 4.2. Abstract actions and abstract plans semi-commute with the cc-operator. Specifically, if $\mathbf{A}$ is an abstract action or an abstract plan, then

$$\mathbf{A}\Big(\overset{cc}{\sum}{}_{i=1}^{n} \Lambda_i w_i \Big) \subseteq \overset{cc}{\sum}{}_{i=1}^{n} \Lambda_i \mathbf{A}(w_i).$$

Proof. An element of the set $\mathbf{A}(\overset{cc}{\sum}{}_{i=1}^{n} \Lambda_i w_i)$ must have the form $A(\overset{cc}{\sum}{}_{i=1}^{n} \lambda_i P_i)$, where $A \in \mathbf{A}, \lambda_i \in \Lambda_i$, and $P_i \in w_i$. Since concrete actions (plans) commute with the cc-operator, this sum can be written as $\overset{cc}{\sum}{}_{i=1}^{n} \lambda_i A(P_i)$, which is clearly an element of $\overset{cc}{\sum}{}_{i=1}^{n} \Lambda_i \mathbf{A}(w_i)$. $\square$

As a consequence of the above theorem, if the initial world w_{ini} is a set of probability distributions [3] that can be written as $w_{ini} = \overset{cc}{\sum} \Lambda_i w_i, w_i \subseteq \mathcal{S}$, then the expected utility interval of an abstract plan $\mathbf{pl}$, projected on w_{ini} can be approximated as:

[3] We use the term "world" to describe the state of affairs after projecting several abstract actions. Since abstract actions are mappings that return *sets of probability distributions*, a world now can be a set of probability distributions.

$$
\begin{aligned}
\mathrm{conv}(u(\mathbf{pl}(w_{ini}))) &= u(\mathrm{conv}(\mathbf{pl}(w_{ini}))) \\
&= u(\mathrm{conv}(\mathbf{pl}(\overset{cc}{\sum} \Lambda_i w_i))) \\
&\subseteq u(\mathrm{conv}(\overset{cc}{\sum} \Lambda_i \mathbf{pl}(w_i))) \text{ } (\mathbf{pl} \text{ semi-commutes with cc-op}) \\
&= \overset{icc}{\sum} \Lambda_i u(\mathrm{conv}(\mathbf{pl}(w_i))) \text{ (conv and } u \text{ commute with cc-op)},
\end{aligned}
$$

where the first equality is due to the fact that utility functions and convex hull operator also commute, i.e. $\mathrm{conv}(u(K)) = u(\mathrm{conv}(K))$ for all $K \subseteq \mathcal{S}$. These equalities illustrate the idea of switching from probability numbers to probability intervals, in the context of projecting and evaluating abstract plans, by replacing the probability cross-product with the cc-operator. The interesting interplay among abstract actions/plans, utility functions, convex hull, and the cc-operator plays a key role in projecting and evaluating abstract probabilistic plans in the DRIPS planner. This important property allows the process of evaluating abstract plans to be performed in two steps. In the first step, we project each branch of each action in the plan on each branch of the current world, expanding a *pcc-tree* in the process (a world is now a set of probability distributions represented by a pcc-tree). In the second step, we compute the expected utility interval of the plan by replacing the leaves of the final pcc-tree with their utility values and evaluating the resulting *icc-tree*. The reader is refered to [12] for a more thorough discussion of various projection algorithms for DRIPS.

The next theorem states that the convex hull of projecting an abstract action/plan on a world can be computed by projecting on the convex hull of that world. This result is heavily utilized by DRIPS, since it is often convenient to approximate a set of probability distributions by its convex hull.

Theorem 4.3. Suppose $w \subseteq \mathcal{S}$, and $\mathbf{A}$ is an abstract action or an abstract plan. Then $\mathrm{conv}(\mathbf{A}(w)) = \mathrm{conv}(\mathbf{A}(\mathrm{conv}(w)))$.

Proof. The "$\subseteq$" relation is obvious, so we only need to prove the "$\supseteq$" relation, which amounts to proving that $\mathbf{A}(\mathrm{conv}(w)) \subseteq \mathrm{conv}(\mathbf{A}(w))$. An element of $\mathbf{A}(\mathrm{conv}(w))$ must have the form $A(\overset{cc}{\sum}_{i=1}^{n} \lambda_i P_i)$, where $A \in \mathbf{A}$ and $P_i \in w$. Since concrete actions/plans commute with the cc-operator, this cc-sum can be written as $\overset{cc}{\sum}_{i=1}^{n} \lambda_i A(P_i)$, which is clearly an element of $\mathrm{conv}(\mathbf{A}(w))$. $\square$

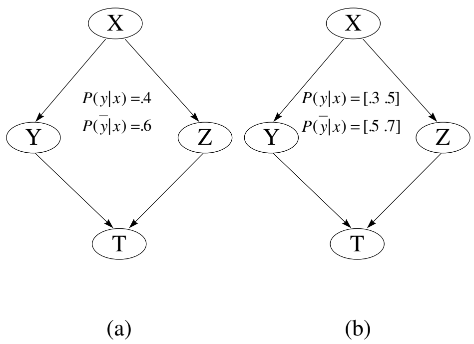

Figure 5. Examples: (a) a Bayesian network and (b) an interval Bayesian network.

5. INTERVAL BAYESIAN NETWORKS

In this section we demostrate how the cc-operator can be used to make inference in interval Bayesian networks.

A Bayesian Network is a pair consisting of a directed acyclic graph, and a collection of probability tables associated with the nodes. Each node of the graph represents a random variable, and its associated probability table specifies the conditional probability of that node given its parents. A Bayesian network represents a joint probability distribution over the random variables corresponding to the nodes, and this distribution is obtained by multiplying the conditional probability tables. Figure 5.a depicts a 4-node Bayesian network N, where the conditional probabilities to be specified are $P(X), P(Y|X), P(Z|X)$, and $P(T|Y, Z)$ (for example, $P(y|x) = .4$, and $P(\bar{y}|x) = .6$). This Bayesian network represents the joint probability distribution P over $\{X, Y, Z, T\}$: $P(X, Y, Z, T) = P(X)P(Y|X)P(Z|X)P(T|Y, Z)$).

Interval Bayesian networks (IBN) [10,25,8] are a generalization of Bayesian networks where we allow the probabilities to be interval-values. An IBN represents a set of Bayesian networks, each of which is obtained from the IBN by *instantiation*, a process that replaces each interval probability with a *consistent* point probability, where consistency means that the points must be in their corresponding intervals, and obey the axioms of probability. Figure 5.b depicts a 4-node interval Bayesian network $\mathbf{N}$ where $\mathbf{P}(y|x) = [.3\ .5]$ and $\mathbf{P}(\bar{y}|x) = [.5\ .7]$. The regular Bayesian network in Figure 5.a (N) is an instantiation of $\mathbf{N}$. We denote the instantiation relationship by $N \in \mathbf{N}$.

Given such an IBN $\mathbf{N}$, the answer to any probabilistic query is defined to be an interval whose endpoints are the infimum and supremum of that query, ranging over all Bayesian network instantiations of $\mathbf{N}$. For example, in the IBN in Figure 5.b, we might be interested in computing $\mathbf{P}(t)$, which is defined as $[\inf_{N\in\mathbf{N}} P_N(t)\ \sup_{N\in\mathbf{N}} P_N(t)]$.

Dechter [6] showed that most of Bayesian network algorithms for updating probabilities, finding the most probable explanation, finding the maximum a posteriori hypothesis, and computing maximum expected utility, can be cast into a unifying algorithm schema called *bucket elimination*. We now present an interval version of this schema and that computes the bounds for the probability of any node in an IBN. In this section we consider the case when there is no evidence in the IBN. The problem of answering probabilistic queries in IBNs with evidence will be studied in the Section 7 where we discuss evidential updating with pcc-trees.

The main idea of this algorithm can be best understood through an example. Suppose that in the IBN $\mathbf{N}$ in Figure 5.b, we are interested in computing the probability bounds for the random variable T taking on some value t. Suppose that P is the joint distribution represented by a BN instantiation N of $\mathbf{N}$. Note that $P(t)$ can be written as $P(t) = \sum_{X,Y,Z} P(X,Y,Z,t) = \sum_{X,Y,Z} P(X)P(Y|X)P(Z|X)P(t|Y,Z)$. Our goal is to factorize this last sum into a series of probability cross-products, which can be generalized into a series of cc-operators. For example, we have that:

$$\begin{aligned} P(t) &= \sum_{X,Y,Z} P(X)P(Y|X)P(Z|X)P(t|Y,Z) \\ &= \sum_{X} P(X) \sum_{Y} P(Y|X) \sum_{Z} P(Z|X)P(t|Y,Z). \end{aligned}$$

This factorization corresponds exactly to what we get by applying the bucket elimination algorithm to the ordering $\{X,Y,Z,T\}$ of the random variables. From the fact that $P(X) \in \mathbf{P}(X), \sum_X P(X) = 1$, we can see that this formula can be generalized into a cc-operator sum. Now, given that X is fixed at some value x, we have that $P(Y|x) \in \mathbf{P}(Y|x), \sum_Y P(Y|x) = 1$, which means that we can generalized the inside sum into a cc-operator sum. It is obvious that all of the remaining sums in this formula can be generalized into cc-operator sums. Thus we have:

$$
\begin{aligned}
P(t) &= \sum_{X,Y,Z} P(X)P(Y|X)P(Z|X)P(t|Y,Z) \\
&= \sum_{X} P(X) \sum_{Y} P(Y|X) \sum_{Z} P(Z|X)P(t|Y,Z) \\
&\in \overset{icc}{\sum}{}_X \mathbf{P}(X) \ \overset{icc}{\sum}{}_Y \mathbf{P}(Y|X) \ \overset{icc}{\sum}{}_Z \mathbf{P}(Z|X)\mathbf{P}(t|Y,Z).
\end{aligned}
$$

The algorithm for the general case is the following. Suppose that we are interested in computing the bounds of the probability $\mathbf{P}(X)$ for some node X in an IBN $\mathbf{N}$. Let $\{X_1, X_2, \ldots, X_n\}$ be *any* topological (or ancestral) ordering of the random variables of $\mathbf{N}$, and $X = X_k$. Let $\pi_i, i = 1, \ldots, k$ denote the set of parents of node X_i. Then, the bounds of $\mathbf{P}(X_k = x_k)$ can be approximated as:

$$
\begin{aligned}
\mathbf{P}(x_k) &= \sum_{X_1,\ldots,X_{k-1}} \mathbf{P}(X_1, \ldots, X_{k-1}, x_k) = \sum_{X_1,\ldots,X_{k-1}} \prod_{i=1}^{k} \mathbf{P}(X_i|\pi_i) \\
&\subseteq \overset{icc}{\sum}{}_{X_1} \mathbf{P}(X_1) \ \overset{icc}{\sum}{}_{X_2} \mathbf{P}(X_2|\pi_2) \ \overset{icc}{\sum} {}_{X_3} \cdots \ \overset{icc}{\sum}{}_{X_{k-1}} \mathbf{P}(X_{k-1}|\pi_{k-1})\mathbf{P}(x_k|\pi_k).
\end{aligned}
$$

Note that the bounds for $\mathbf{P}(x_k)$ obtained by this algorithm are correct, but not tight. The same can be said of the algorithm suggested in Theorem 4.2 for projecting and evaluating abstract plans. What's more, the two problems (of evaluating abstract plans and answering IBN quesries) look eerily similar, and a closer look at the two problems reveals that the latter is in fact a special case of the former: answering an IBN query can be viewed as an instance of abstract plan evaluation with suitable choices of worlds and actions. The precision loss of the algorithm for evaluating abstract plans is due to the fact that although the icc-operator sum – when applied only once – yields exact results, computations of globally related icc-operators are performed *locally* and *independently.* (An analogue of this loss occurs when we approximate the maximization of the sum of *correlated* individuals by summing the maximization of the individuals.) This is the precision tradeoff we have to make if we are to avoid having to resort to computationally more costly techniques such as linear programming. Experiments with the DRIPS planner show that the icc-tree approximation produces sufficiently tight bounds in order to eliminate large classes of suboptimal plans.

6. PCC-TREES AS A REPRESENTATION FOR SETS OF PROBABILITY DISTRIBUTIONS

In this section, we investigate pcc-trees as data structures that encode sets of probability distributions. We first show that the class of pcc-trees is exactly the class of polytope-like sets of probability distributions, and discuss the complexity of switching between these two representations. We then show that Dempster-Shafer belief functions, when interpreted as lower envelopes of sets of distributions, can be represented by 2-level pcc-trees in at least two different ways.

6.1. PCC-TREES AND POLYTOPE-LIKE SETS OF PROBABILITY DISTRIBUTIONS

One of the first question we need to answer when introducing pcc-trees is: "What kind of sets of probability distributions do pcc-trees encode?". It follows from Theorem 2.2 (the cc-operator and convex hull operator commute) that any pcc-tree is equal to its convex hull and thus encodes a convex set of probability distributions. Also recall that Figure 2.b demonstrates that any polytope-like set of probability distributions can be represented by a 2-level pcc-tree where the nodes at level 1 are the vertices of the polytope (this pcc-tree has a total of at most $1+n+nd$ nodes, where n is the number of the vertices of the polytope, and $d = |\Omega|$). Our next result states that the converse is also true: the cc-operator is closed on the set of polytopes, and thus any pcc-tree represents a polytope-like set of probability distributions. In other words, the class of pcc-trees is precisely the class of polytope-like sets of probability distributions.

Theorem 6.1. If $\Lambda_i \subseteq [0\ 1]$ are closed intervals and $w_i \subseteq \Re^d$ are polytopes, then $K = \vec{\Lambda} \otimes \vec{w} = \overset{cc}{\sum}_{i=1}^{n} \Lambda_i w_i$ is also a polytope.

Proof. We need the following auxiliary lemma, which is a standard result in convex geometry. (See, for example [11].)

Lemma. Suppose that $K \subseteq \Re^d$ is a convex set and $V \subseteq K$ is a finite set of points in K such that for any point $u \in \Re^d$, there is a point $v \in V$ such that $\langle v, u \rangle = \max_{x \in K} \langle x, u \rangle$. Then $K = \text{conv}(V)$, or, equivalently, K is a polytope whose vertices are contained in V.

Now consider any point $u \in \Re^d$. The set of real numbers that are inner products of u with the elements of K can be written as $\langle u, K\rangle = \langle u, \overset{cc}{\sum}_{i=1}^{n} \Lambda_i w_i\rangle = \overset{icc}{\sum}_{i=1}^{n} \Lambda_i \langle u, w_i\rangle$. (In the case when $K, w_i \subseteq \mathcal{S}$, we can view the set K as a set of probability distributions, u as a utility function, and $\langle u, K\rangle$ as the expected utility interval of an abstract plan whose final world is K.) It is easy to see that in order to compute the bounds of the interval $\langle u, w_i\rangle$, we only have to minimize and maximize $\langle u, x\rangle$ over the *finite* set of vertices of the polytope w_i. From this and the fact that the weights λ_i (chosen by the greedy knapsack algorithm for evaluating icc-operator sum, see Section 3) are chosen based only on the *order* of the values of the material (there are $n!$, i.e. *finite* number of possible orders), it follows that there are a finite set of points in K that satisfies the condition of the auxiliary Lemma. □

It is not an easy task to determine the vertices of a polytope represented by a pcc-tree. It is however possible to estimate the number of vertices of a polytope represented as a pcc-tree. We have found that the number of vertices of a polytope represented as a pcc-tree with n nodes is asymptotically bounded above by $O((2e^{1/e})^n)$. The proof of this result is rather involved and is omitted here. We can obtain a very rough upper bound using the following argument. Denote by $v(K)$ the number of vertices of a polytope K. Denote by $\mathcal{PCC}(n)$ the set of polytopes representable by pcc-trees of at most n nodes, and $v(n) = \sup_{K \in \mathcal{PCC}(n)} v(K)$. Now assume that $K = \overset{cc}{\sum}_{i=1}^{m} \Lambda_i w_i, 2 \leq m \leq n$ is a polytope represented by a pcc-tree with $(n+1)$ nodes, where w_i are also polytopes. Since w_i are polytopes representable with pcc-trees of at most n (in fact $n-1$, since $m \geq 2$) nodes, we have that $v(w_i) \leq v(n)$. The proof of Theorem 6.1 implies that $v(K) \leq m!(v(n))^m \leq n!(v(n))^n$. Since K was chosen arbitrarily, we have that $v(n+1) \leq n!(v(n))^n$.

Although the recursive upper bound obtained above is clearly very loose, it has an implication that there are polytopes (with sufficiently large number of vertices) that require pcc-trees with arbitrarily large number of nodes to represent them. Another implication of this bound is the following theorem.

Theorem 6.2. For all positive integers n, $\mathcal{PCC}(n)$ is a strict subset of $\mathcal{PCC}(n+2)$.

Proof. Let $K \in \mathcal{PCC}(n)$ and $v(K) = v(n)$. Such K exists since $v(n) < \infty$. We will construct from K a polytope K' representable by a pcc-tree with $n+2$ nodes which does not belong to $\mathcal{PCC}(n)$.

Fix a point $s \in \mathcal{S} - K$ (such a point must exist unless K is the entire probability simplex). Let $I = [0\ 1/2]$, $J = [1/2\ 1]$, and $K' = (I, J) \otimes (\{s\}, K)$. Also set $K(s, 1/2) = \{1/2(s + x) | x \in K\}$. It is easy to verify that 1) $K' = \text{conv}(K(s, 1/2) \cup K)$ and 2) $v(\text{conv}(K(s, 1/2) \cup K)) > v(K)$, which establish the theorem. □

6.2. PCC-TREES AND DEMPSTER-SHAFER BELIEF FUNCTIONS

In this subsection we show that any Dempster-Shafer belief function [7, 20], when viewed as a lower envelope of a set of probability distributions, can be represented by a 2-level pcc-tree, using its corresponding mass assignment function. We first need some preliminaries to formulate this result.

Let Ω be a finite set [4]. A function $Bel : 2^{\Omega} \to [0, 1]$ is called a *belief function* if $Bel(\emptyset) = 0$, $Bel(\Omega) = 1$, $Bel(A) \leq Bel(B)$ whenever $A \subseteq B$, and for all $A_i \subseteq \Omega, i = 1, \ldots, k$:

$$Bel\left(\bigcup_{i=1}^{k} A_i\right) \geq \sum_{\emptyset \neq I \subseteq \{1,\ldots,k\}} (-1)^{|I|+1} Bel\left(\bigcap_{i \in I} A_i\right).$$

The corresponding *mass assignment function* [5] $m : 2^{\Omega} \to [0, 1]$ of a belief function Bel is obtained by the Möbius transformation as $m(A) = \sum_{B \subseteq A} (-1)^{|A-B|} Bel(B), \forall A \subseteq \Omega$. It is well-known that $\sum_{A \subseteq \Omega} m(A) = 1$. The subsets $A \subseteq \Omega$ that are assigned positive masses ($m(A) > 0$) are called the *focal elements* of the belief function Bel. A belief function Bel can also be viewed as the lower envelope of a set of probability distributions *consistent* with it, where a probability distribution P is consistent with Bel if $Bel(A) \leq \sum_{s \in A} P(s), \forall A \subseteq \Omega$.

Kyburg [16] showed that the set of probability distributions consistent with a belief function is closed and convex. Our next result asserts something stronger: this set is a polytope-like set of probability distributions, which is equal to the weighted sum of the faces of the probability simplex, with faces $\text{conv}(A), A \subseteq \Omega$ weighted with $m(A)$.

Theorem 6.3. $Bel = \overset{cc}{\sum}_{i=1}^{k} m(A_i)\text{conv}(A_i)$. Here both sides are interpreted as sets of probability distributions (See Figure 2.c for an example).

[4] Ω is often called *frame of discerment.*
[5] Also called *basic probability assignment.*

Proof. It is well-known [24,7,2] that a probability distribution P is consistent with Bel if and only if there exist so-called *allocating functions* $g_i : \Omega \rightarrow [0,1], i = 1,\ldots,k$ such that:

1. For all $i = 1,\ldots,k$ and $s \in \Omega$, $g_i(s) = 0$ if $s \notin A_i$.
2. $\sum_{s\in\Omega} g_i(s) = \sum_{s\in A_i} g_i(s) = m(A_i)$.
3. $\sum_{i=1}^{k} g_i(s) = P(s)$.

The theorem is established by the equality: $P = \overset{cc}{\sum}_{i=1}^{k} m(A_i)\frac{g_i}{m(A_i)}$. □

7. EVIDENTIAL UPDATING WITH PCC-TREES

Theorem 6.1 raises a natural question: If the class of pcc-trees is identical to the class of polytope-like sets of probability distributions, why shoud not we just use well-known methods of representing polytopes such as vertex-listing, or linear programming? Given a pcc-tree, we can in principle either determine the vertices of the polytope it represents, or establish a system of linear inequalities that characterizes this polytope. In the abstraction-based probabilistic planning framework described in Section 4, neither of these two approaches appears to be very efficient. For example, if we represent uncertainty of the world by the list of vertices of a polytope, we need to compute this list after each action projection, which is a non-trivial task. Similarly, it seems no less difficult to determine a system of linear inequalities that characterizes uncertainty of the world after each action projection. Using pcc-trees, the plan projection process reduces to a simple tree-spanning process, and the plan evaluation (expected utility computation) process reduces to a simple icc-tree evaluation process. Furthermore, the cc-operator and pcc-trees have a number of properties that are particularly useful for the purpose of abstracting actions and plans ([12], Lemma 1-4).

However, in the context of evidential updating, the pcc-tree representation turns out to be less suitable than its vertex-listing and linear programming counterparts. We investigate the reasons of this limitation of pcc-trees in this section.

The Bayesian approach to dealing with uncertainty involves representing the uncertain information with a *prior* probability distribution $P(.)$, and updating it with an observed evidence E to obtain a *posterior* probability distribution $P(.|E)$, which is defined as $P(x|E) = P(x,E)/P(E)$. Following [9], we view an evidence E as a function, called *evidential function* that maps probability functions to probability functions: $E : P \mapsto E(P) = P(.|E)$. For probability functions P where $P(E) = 0$, $E(P)$ is undefined. In this paper we consider only *certain* (also called *hard*) evidence E that is a subset of the sample space: $E \subseteq \Omega$.

The Bayesian paradigm is often challenged due to the its requirement that a *single* prior probability distribution be specified, even when little statistical information is available. The *cautious Bayesian* approach addresses this issue by taking into consideration a (usually convex) set K of prior probability distributions, and updating each member $P \in K$ of this set with the observed evidence E to obtain $E(K) = \{E(P)|P \in K, P(E) > 0\}$. This evidential updating approach is refered to as *convex conditionalization* [18,16].

Convex conditionalization for various representations of convex sets of probability distributions has been a subject of extensive study. White [28] studied the case where both the prior set of distributions and the (uncertain, also called *soft*) evidence are characterized by systems of linear inequalities. Snow [22] showed that in the case where the evidence is a single uniformly positive distribution, White's method of conditioning yields exact results.

Convex conditionalization has also been studied in the case where the set K of prior distributions is a set of distributions consistent with a Dempster-Shafer belief function Bel. Fagin and Halpern [9] showed that the lower and upper envelopes of $E(Bel)$ correspond to a pair of belief (denoted $Bel(.|E)$) and plausibility functions. Jaffray [15] showed that $E(Bel)$ in general is a *strict* subset of the set of distributions consistent with the new belief function $Bel(.|E)$. This result means that the belief function representation loses information in the updating process. Chrisman [3] showed that this loss of information is especially troublesome when pieces of evidence are to be updated incrementally, since different orders of pieces of evidence produce different results (which are also different from the result obtained when updating all the pieces of evidence in a single step). By using Dempster conditioning bounds in conjunct with the convex conditioning bounds, Chrisman was able to develope an exact conditioning procedure for 2-monotone lower probabilities, the class of which contains the class of belief functions [3].

We now investigate the problem of convex conditioning on pcc-trees. Suppose that $K = \overset{cc}{\sum}{}_{i=1}^{n}\Lambda_i w_i$ is a pcc-trees where w_i are also pcc-trees that are subtrees of K. Suppose that $E \subseteq \Omega$ is hard evidence. We are interested in determining $E(K) = E(\overset{cc}{\sum}{}_{i=1}^{n}\Lambda_i w_i)$. We start out by studying this problem in special cases of pcc-trees.

In the case when $K = \text{conv}(A), A \subseteq \Omega$ is a face of the probability simplex (Figure 2.a), then clearly $E(K) = E(\text{conv}(A)) = \text{conv}(E \cap A)$. In the more general case when $K = \text{conv}\{P_1, \ldots, P_k\}$ is a polytope-like sets of probability distributions with vertices P_i, it can be shown that $E(K)$ is also a polytope whose vertices are contained in the set $\{E(P_i) | i = 1, \ldots, k\}$ [21,23]. Below we provide a proof of this statement that is similar to the proof given in [23], since it is highly instructive in our upcoming analysis.

Theorem 7.1. $E(\text{conv}\{P_1, \ldots, P_k\}) = \text{conv}(E(P_1), \ldots, E(P_k)\})$.

Proof. Like most arguments involving convex combinations, here we need to consider only the case of $k = 2$. The case when either $P_1(E) = 0$ or $P_2(E) = 0$ is easy to check, so we suppose that $P_1(E) > 0, P_2(E) > 0$. An element of the left-hand side has the form $E(\alpha P_1 + (1-\alpha)P_2), \alpha \in [0\ 1]$, while an element of the right-hand side has the form $\beta E(P_1) + (1-\beta)E(P_2), \beta \in [0\ 1]$. The set equality is established if we can construct a one-to-one mapping from any $\alpha \in [0\ 1]$ to a $\beta \in [0\ 1]$ such that $E(\alpha P_1 + (1-\alpha)P_2) = \beta E(P_1) + (1-\beta)E(P_2)$. It is easy to check that the mapping $\beta = \alpha/[\alpha + (1-\alpha)P_2(E)/P_1(E)]$ satisfies these requirements. $\square$

An immediate observation from the above proof of Theorem 7.1 is that the evidential function E does not preserve convex combinations and thus is *not* a convex mapping [6]. As one would suspect, this non-convexity of the evidential function causes difficulty in updating pcc-trees, since the evidential function does not commute with the cc-operator.

An example that illustrates this point is the case of updating belief functions. Recall that a belief function Bel with its corresponding mass assignment function m and focal elements $A_1, \ldots, A_k$ can be written as $Bel = \overset{cc}{\sum}{}_{i=1}^{k} m(A_i)\text{conv}(A_i)$ (see Theorem 6.3 and Figure 2.c). If the evidential function E had commuted with the cc-operator, we would have obtained $E(Bel) = E(\overset{cc}{\sum}{}_{i=1}^{k} m(A_i)\text{conv}(A_i)) = \overset{cc}{\sum}{}_{i=1}^{k} m(A_i)E(\text{conv}(A_i)) = \overset{cc}{\sum}{}_{i=1}^{k} m(A_i)\text{conv}(A_i \cap E)$, which is identical to $Bel \oplus E$, the result of combining the belief function Bel with the evidence E using

[6] Further investigation based on geometric arguements shows that this phenomenon occurs because of the *normalization step* of updating, when the probabilities $P(x, E)$ are normalized by dividing by $P(E)$ so that they sum up to 1.

Dempster's rule of combination. This is impossible since it is well-known that the bounds obtained using Dempster's rule in general are *strictly narrower* than the bounds obtained using convex conditionalization [7,9].

The difficulty of evidential updating with pcc-trees translates into the difficulty of answering probabilistic queries in interval Bayesian networks that have evidence. A straightforward attempt to extend the algorithm described in Section 5 to IBNs with evidence involves maintaining two different representations of polytope-like sets of probability distributions: with a pcc-tree and with a list of the polytope's vertices. We can use the second representation to incorporate evidence according to Theorem 7.1, and use the first representation to compute the bounds for probabilistic queries. Effective realization of this idea requires efficient algorithms for alternating between the two representations. We leave this as an issue for future research.

References

[1] C. Boutilier and R. Dearden. Using abstractions for decision-theoretic planning with time constraints. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 1016–1022, Seattle, July 1994.

[2] L. Chrisman. Abstract probabilistic modeling of action. In *Proceedings of the First International Conference on Artificial Intelligence Planning Systems*, pages 28–36, June 1992.

[3] L. Chrisman. Incremental conditioning of lower and upper probabilities. *International Journal of Approximate Reasoning*, 13(1):1–25, 1995.

[4] F. Cozman. Robustness analysis of bayesian networks with local convex sets of distributions. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 108–115, August 1997.

[5] T. Dean, L. Pack Kaelbling, J. Kirman, and A. Nicholson. Planning under time constraints in stochastic domains. *Artificial Intelligence*, 76(1-2):35–74, July 1995.

[6] R. Dechter. Bucket elimination: A unifying framework for probabilistic inference algorithms. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 211–219, August 1996.

[7] A. Dempster. Upper and lower probabilities induced by a multi-valued mapping. *Annals of Mathematical Statistics*, 38(2):325–339, 1967.

[8] D. Draper. *Localized Partial Evaluation of Bayesian Belief Networks*. PhD thesis, Computer Science Dept., Univ. of Washington, Seattle, November 1995.

[9] R. Fagin and J. Halpern. A new approach to updating belief. In P. Bonissone, M. Henrion, L. Kanal, and J. Lemmer, editors, *Uncertainty in Artificial Intelligence 6*, pages 347–374. 1991.

[10] K.W. Fertig and J.S. Breese. Probability intervals over influence diagrams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):280–286, March 1993.

[11] B. Grunbaum. *Convex Polytopes.* Interscience, London, New York, 1967.

[12] V. Ha and P. Haddawy. Theoretical foundations for abstraction-based probabilistic planning. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 291–298, Portland, August 1996.

[13] P. Haddawy, A. Doan, and C.E. Kahn Jr. Decision-theoretic refinement planning in medical decision making: Management of acute deep venous thrombosis. *Medical Decision Making*, 16(4):315–325, Oct/Dec 1996.

[14] S. Hanks. Practical temporal projection. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 158–163, Boston, July 1990.

[15] J. Jaffray. Bayesian updating and belief functions. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5):1144–1152, 1992.

[16] H. Kyburg Jr. Bayesian and non-bayesian evidential updating. *Artificial Intelligence*, 31(3):271–294, 1987.

[17] H. Kyburg and M. Pittarelli. Set-based bayesianism. *IEEE Transactions on Systems, Man, and Cybernetics*, 26:324–339, 1996.

[18] I. Levi. *The Enterprise of Knowledge.* MIT Press, Cambridge,MA, 1980.

[19] T. Seidenfeld, M. Schervish, and J. B. Kadane. On the shared preferences of two bayesian decision makers. *Journal of Philosophy*, 86:225–244, 1989.

[20] G. Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, Princeton, NJ, 1976.

[21] P. Snow. Bayesian inference without point estimates. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 233–237, 1986.

[22] P. Snow. Improved posteriori probability estimates from prior and conditional linear constraint systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(2):464–469, 1991.

[23] W. Stirling and A. Morrell. Convex bayesiandecision theory. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(1):173–183, 1991.

[24] V. Strassen. Messfehler and information. *Z. fur Wahrscheinlichkeits-theorie und Verw. Gebiete*, 2:273–305, 1964.

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

[26] S. Thiebaux, J. Hertzberg, W. Shoaff, and M. Schneider. A stochastic model of actions and plans for anytime planning under uncertainty. *International Journal of Intelligent Systems*, 1994.

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

[28] C. White. A posteriori re[resentations based on linear inequality descriptions of a priori and conditional probabilities. *IEEE Transactions on Systems, Man, and Cybernetics*, 16:570–573, 1986.