Queen's University
Belfast

Partially Ordered Preferences in Decision Trees: Computing Strategies with Imprecision in Probabilities

Published in:
Proceedings of the IJCAI Workshop about Advances on Preference Handling

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Partially Ordered Preferences in Decision Trees:
Computing Strategies with Imprecision in Probabilities

Daniel Kikuti
Escola Politécnica
University of São Paulo
daniel.kikuti@poli.usp.br

Fabio G. Cozman
Escola Politécnica
University of São Paulo
fgcozman@usp.br

Cassio P. de Campos
Pontifícia Universidade Católica
PUC-SP
cassio@pucsp.br

Abstract

Partially ordered preferences generally lead to choices that do not abide by standard expected utility guidelines; often such preferences are revealed by imprecision in probability values. We investigate five criteria for strategy selection in decision trees with imprecision in probabilities: "extensive" Γ-maximin and Γ-maximax, interval dominance, maximality and E-admissibility. We present algorithms that generate strategies for all these criteria; our main contribution is an algorithm for E-admissibility that runs over admissible strategies rather than over sets of probability distributions.

1 Introduction

A rational agent is often expected to comply with strict guidelines concerning decisions: acts are encoded as functions from states to consequences, consequences are measured by utilities, and utilities are weighted by probabilities. The rational agent is then assumed to have a complete order that ranks all acts: any two decisions can be compared, and either one is better than the other, or the two are equivalent. Preferences are then revealed by the agent's consistent pattern of choice among acts [Samuelson, 1948].

In this paper we wish to explore situations where preferences are partially ordered: given two acts, the agent may prefer one to the other, or find them to be equivalent, *or find them to be incomparable*. In this paper we want to restrict attention to models that assume a *unique* utility function (up to a linear transformation) but that contemplate *non-unique* probability values as a source of partially ordered preferences [Jaffray, 1999; Machina, 1989; Seidenfeld, 2004; Walley, 1991]. Imprecise beliefs may arise from an incomplete understanding of a decision situation, from lack of prior knowledge or empirical data, from disagreements between experts, or from lack of resources for a complete elicitation procedure [Walley, 1991]. A gradual assessment of preferences may in fact create intermediate models that are incomplete but that are still useful for decisions [Wang and Boutilier, 2003]. Whatever may be its origin, imprecise beliefs represented by a set of probability measures lead to partially ordered preferences because each probability measure in the set may create a different complete ordering among acts: the intersection of these complete ordering is the agent's partial ordering.

Thus our agents express their partially ordered preferences by precise utilities and imprecise beliefs. Consider first the *static* scenario where the agent must select a single act. The agent may select an act that maximizes the minimum expected utility. This solution is called a Γ-maximin one [Berger, 1985; Gilboa and Schmeidler, 1989]. An alternative solution is to find a *set* of acts such that any act in the set is an optimal act with respect to at least a probability measure in the set of possible probability measures. Such acts are called *E-admissible* [Levi, 1980]. Other solutions, such as *maximality* and *interval dominance*, can be found in the literature, and there is considerable debate on which solution should be adopted in practice [Seidenfeld, 2004; Troffaes, 2004].

In this paper we focus on the more complex *dynamic* situation, where a *decision tree* represents a sequential decision problem with imprecise probabilities. We present algorithms for computing strategies under "extensive" Γ-maximin and Γ-maximax, interval dominance, maximality and E-admissibility — we develop these algorithms within a multilinear programming framework. Our main contribution is an algorithm for E-admissibility whose complexity depends essentially on the number of admissible strategies — thus avoiding a direct dependency on the potentially high complexity of the underlying set of probabilities.

Section 2 briefly reviews the basics of decision trees and sets of probabilities. Section 3 presents several algorithms that handle decision trees from different perspectives. Section 4 presents examples that illustrate our methods, and Section 5 concludes the paper.

2 Decision trees and credal sets

A *decision tree* represents a sequential decision problem using nodes (choice, chance and value nodes) and arcs between nodes [Raiffa, 1968]. Arcs indicate possible decisions (when coming out of choice nodes) or possible states (when coming out of chance nodes). A chance node is associated with probability values, and value nodes are associated with utility values.

An obvious solution method for solving a standard decision tree is by complete enumeration of all strategies (equivalent to the representation of games in normal form [Luce
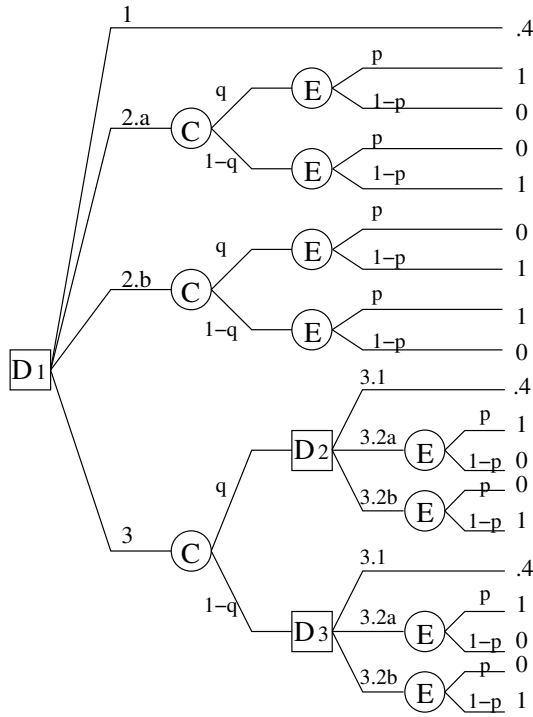
Figure 1: Decision tree for Example 1.

and Raiffa, 1957]). Explicit enumeration clearly becomes infeasible for large problems. A better approach is to take the principle of *dynamic feasibility*[1]: to assess sub-strategy s at a choice node $n-1$, one has to anticipate how one will choose at (potential) "future" choice nodes n, and declare infeasible all future alternatives under s which are inadmissible at n [Seidenfeld, 1988].

Example 1 Figure 1 depicts a sequential decision problem, adapted from [Seidenfeld, 2004]. The chance node C is the toss of a fair coin with probability of 0.5 for heads or tails ($q = 1 - q = 0.5$). Suppose that p is a precise probability value, say 0.25. At the initial choice node, there are four options. We must pay 0.4 utiles to take 1, 2a or 2b, and we pay 0.35 utiles to take decision 3. At D_1 we have twelve strategies to evaluate. Using dynamic feasibility, we only have to evaluate four strategies (1, 2a, 2b,(3, 3.b, 3.a)).

In this paper we *only consider methods that employ dynamic feasibility*. Some solutions, such as Γ-maximin, produce strategies under dynamic feasibility that are not identical to strategies produced in the normal form (that is, when all strategies are enumerated) [Seidenfeld, 2004]. We focus on dynamic feasibility solutions for computational reasons: it is impractical to enumerate all strategies as decision trees grow larger. That is, we always work with the "extensive" form of decision trees.

---

[1]This principle is also known as *backward induction* or is simply taken as the basis for *dynamic programming*.

In this paper we wish to study situations where a chance node is not associated with a single probability measure, but rather with intervals or sets of measures. Such representations encode partially-ordered preferences [Walley, 1991; Seidenfeld, 1995]; here we briefly review essential concepts concerning sets of probabilities.

A *credal set* $K(X)$ is a set of probability distributions (or measures) for random variable X [Levi, 1980]. A credal set captures imprecision in probability values; given a credal set and a function $f(X)$, one may compute *lower expectations* $\underline{E}[f(X)] = \inf E[f(X)]$ and *upper expectations* $\overline{E}[f(X)] = \sup E[f(X)]$, where $E[f(X)]$ denotes standard expectation. Lower and upper probabilities are defined similarly [Giron and Rios, 1980; Walley, 1991]. A conditional credal set is obtained by applying Bayes rule to every distribution in a credal set. We adopt the following definition of independence, usually referred to as *strong independence*: two variables X and Y are strongly independent when the credal set $K(X,Y)$ has all vertices satisfying stochastic independence of X and Y (that is, all vertices factorize as $P(X)P(Y)$) [Couso *et al.*, 2000; Cozman, 2000].

In this paper we assume that all variables are categorical. We also assume that credal sets are closed and convex with finitely many vertices. Finally, we assume that any conditioning event has lower probability strictly larger than zero.

3 Algorithms for decision trees associated with credal sets

In this section we present algorithms that produce one or several strategies for a given decision tree associated with imprecise probabilities — that is, chance nodes are associated with credal sets. We start with the relatively simple "extensive" Γ-maximin and Γ-maximax solutions (here "extensive" indicates that solutions are not necessarily valid for the normal form, but they are valid in an extensive form employing dynamic feasibility). We then consider interval dominance, maximality and E-admissibility solutions — all of which typically produce sets of strategies [Troffaes, 2004].

Given our reliance on dynamic feasibility, all solutions follow a iterative plan: start with the leaves of the tree, and gradually build the partial strategies that are admissible from a certain point on. The skeleton of the procedure is as follows:

DECISIONTREE(*decisiontree*)
1   **for** each decision node D from depth N to 1 **do**
2       $Aux \leftarrow$ null;
3   **for** each branch i of D **do**
4   **if** i links D to a choice node **then**
5               $Aux \leftarrow Aux \cup$ STRATEGIES(i);
6   **else if** i links D to a chance node **then**
7               $Aux \leftarrow Aux \cup$ COMBINATION(i);
8   **else** //value node
9               $Aux \leftarrow Aux \cup i$;
10   **endif**
11   **endfor**
12      $X.Adm \leftarrow$ CRITERION–X(Aux);
13 **endfor**
14 **return** STRATEGIES(*root of the decision tree*);

The function STRATEGIES receives a decision node and returns a list of admissible strategies. The function COM-

BINATION receives a chance node and makes recursively the combination of all admissible strategies available on decision nodes and value nodes. These functions simply build lists of admissible strategies as the algorithm proceeds.

The function CRITERION–X is a generic function that must be properly implemented to select all valid strategies in an array of strategies. This function is implemented in several forms in the remainder of this section, replacing X by the appropriated criteria.

Variable Aux is an array of strategies. Each strategy is defined by an array of choices; in Example 1, we have strategies (3, 3.b, 3.b) and (3, 3.a, 3.b), among others.

It is important to understand that a strategy defines a *multilinear* constraint as long as the probabilities that influence the strategy are imprecise. Consider for instance the strategy (3, 3.a, 3.b) in Example 1. If probabilities q and p are interval-valued, then the value of the strategy is the multilinear expression $pq + (1 - p)(1 - q)$. All algorithms presented in this paper require the computation of upper and lower expectations for strategies or algebraic operations on strategies; these upper and lower expectations are obtained by multilinear programming. To compute an upper expectation, it is necessary to maximize a multilinear function subject to whatever constraints are imposed on probability values.

In some problems it may be the case that the necessary probabilities are not directly specified and must be generated through Bayes rule. For example, it may be necessary to manipulate $P(X|Y)$ for variables X and Y, but the decision tree may be associated with probabilities $P(Y|X)$ and $P(X)$ — this is particularly common in influence diagrams. If $P(Y|X)$ and $P(X)$ are precisely specified, direct application of Bayes rule yields $P(X|Y)$. If $P(Y|X)$ or $P(X)$ are only specified up to constraints, then it is necessary to manipulate values of $P(Y|X)$ as unknowns and to introduce the multilinear constraints $P(Y|X)\,P(X) = P(X|Y)\,P(Y)$. Note that this constraint corresponds to Bayes rule.

To solve multilinear programs, we have used Sherali and Tuncbilek's Reformulation-Linearization (RL) method [Sherali, 1992] in our implementation. The RL method substitutes each product of variables $\prod_{j \in J_{rt}} \theta_j$ by a new artificial variable $\vartheta_{J_{rt}}$ for all terms t in the problem, thus obtaining a linear program. The solution of each linear problem gives an upper bound to the solution of the multilinear problem. The method iterates over the variables by branching over their ranges whenever necessary, until each $\vartheta_{J_{rt}}$ is close enough to $\prod_{j \in J_{rt}} \theta_j$. We have used the same method for computation of upper and lower conditional probabilities in multivariate models with remarkable success [Campos and Cozman, 2004].

3.1 Γ-Maximin and Γ-Maximax

The Γ-maximin criterion selects the strategy with highest lower expected value — a "pessimistic" solution [Berger, 1985; Gilboa and Schmeidler, 1989]. In a sequential setting, the "extensive" Γ-maximin solution is to take, at each choice node, the Γ-maximin solution at that point (this may be different from the normal form) [Jaffray, 1999; Seidenfeld, 1995]. The Γ-maximax criterion selects the strategy with highest upper expected value — certainly a very "optimistic" solution

[Satia and Lave, 1973].

This "extensive" form of Γ-maximin leads to a procedure that selects a single strategy for any given set of strategies. Note that there may be several strategies with the same highest lower expected value, but these are all equivalent for this criterion. The resulting algorithm is computationally simple and similar to solution of standard decision trees; the only difference is that the computation of a lower expectation requires multilinear programming.

CRITERION–Γ-MAXIMIN(Aux)
1    $S \leftarrow$ null; $MS \leftarrow -\infty$;
2   **for** each s in Aux **do**
3   **if** $\underline{E}[s] > MS$ **then**
4          $S \leftarrow s; MS \leftarrow \underline{E}[S]$;
5   **endif**
6   **endfor**
7   **return** MS;

The Γ-maximax solution has the same structure, but instead of comparing lower expectations in line 3, we must compare upper expectations. Both the Γ-maximin and the Γ-maximax lead to a *single* strategy, even though the underlying preferences are partially ordered.

3.2 Interval dominance

Interval dominance classifies admissible choices according to a strict partial ordering. The ordering is generated by pairwise comparison. Given two strategies r and s, if $\underline{E}[r] > \overline{E}[s]$, then s is inadmissible. The set of admissible strategies consists of those strategies not classified as inadmissible [Troffaes, 2004]. The algorithm is quite simple, again requiring multilinear programming. In this algorithm we associate an "attribute" *admissible* to each strategy.

CRITERION–INTERVAL DOMINANCE(Aux)
1    $MS \leftarrow$ CRITERION–GAMMA-MAXIMIN(Aux);
2   **for** i running over every strategy in Aux **do**
3   **if** $\underline{E}[MS] > \overline{E}[Aux[i]]$ **then**
4          $Aux[i].admissible \leftarrow$ **false**;
5   **endif**
6   **endfor**
7   **return** All alternatives not marked as false;

This algorithm avoids unnecessary computations of upper and lower expected values: instead of conducting explicit pairwise comparisons, it uses the Γ-maximin solution to yield a linear number of multilinear programs (linear on the number of possible strategies). To show that the algorithm is correct, note that the choice with maximum lower expectation is always admissible according to interval dominance criterion, and the comparison of all strategies with the Γ-maximin strategy is sufficient for determine the admissible ones.

3.3 Maximality

The maximality criterion is also based on pairwise comparisons between strategies. Consider that a credal set represents the imprecise beliefs of a particular problem. A strategy r is *maximal* provided that there is no strategy s such that, for each probability measure P in the credal set, the expected value $E_P[s]$ is larger than $E_P[r]$. The maximality criterion

prescribes that any maximal strategy can be selected by a rational agent; the computational problem is to generate the set of maximal strategies.

CRITERION–MAXIMALITY(Aux)
1   $N \leftarrow$ Number of strategies in Aux;
2   **for** $i = 1 \ldots N\text{-}1$ **do**
3   **for** $j = i\text{+}1 \ldots N$ **do**
4   **if** $\underline{E}[Aux[i] - Aux[j]] > 0$ **then**
5            $Aux[j].admissible =$ **false**;
6   **else if** $\overline{E}[Aux[i] - Aux[j]] < 0$ **then**
7            $Aux[i].admissible =$ **false**;
8   **endif**
9   **endfor**
10 **endfor**
11 **return** All alternatives not marked as false;

The algorithm CRITERION–MAXIMALITY compares any pair of strategies only once. If we know the upper/lower value for $E[s_i - s_j]$, we also know whether one dominates the other and we do not need to evaluate $E[s_j - s_i]$. The term $(s_i - s_j)$ refers to a multilinear expression, obtained by subtracting the multilinear expression of s_j from the multilinear expression of s_i, and of course retaining all constraints on probability values in these expressions. To verify whether N alternatives are admissible, the algorithm runs through at most $\mathcal{O}(N^2)$ multilinear problems.

3.4 E-Admissibility

The criterion of E-Admissibility restricts the decision maker's admissible choices to those that are Bayes for at least one probability measure P in the relevant credal sets. That is, given a choice set S of feasible strategies and a credal set K representing imprecise beliefs, the strategy $s \in S$ is E-admissible when, for at least one $P \in K$, s maximizes expected utility [Schervish *et al.*, 2003]:

$$\mathcal{E} = \bigcup_{p \in K} \arg\max(E[s])$$

If neither option s_i or s_j is E-admissible, then their convex combination $\alpha s_i \oplus (1 - \alpha)s_j$ is not E-admissible. The following fact, similar to the usual decision tree construction, is also true. Let $\mathcal{E}$ be the set of E-admissible strategies and $\neg\mathcal{E}$ the set of E-inadmissible alternatives in a subtree D' of a decision tree D. A strategy $s_i \in \neg\mathcal{E}$ in D' cannot be a substrategy of an E-admissible strategy in D; that is, if we detect that a substrategy is not E-admissible in a subtree, we can discard any strategy that "contains" it.

At first one may think that E-admissibility is qualitatively different from the previous criteria, because it does not directly compare strategies. Rather, E-admissibility looks at distributions on the underlying credal set, and compares strategies for all distributions. Thus one might think that E-admissibility is much more difficult to handle than the previous criteria; in fact this seems to be the existing consensus on the issue [Troffaes, 2004]. However we wish to demonstrate that E-admissibility can also be expressed using pairwise comparisons, when one works in our multilinear programming framework.

For each strategy s, we are interested in finding a probability distribution for which s is optimal in the standard expected utility sense. If this probability distribution exists, then s is E-admissible. That is, strategy $s_i \in S$ is E-admissible if there exists a $P \in K$ such that for all $s_j \in S$, $s_j \neq s_i$, we have $E[s_i - s_j] \geq 0$. These (multilinear) constraints must all be satisfied to show that s_i is E-admissible; if the constraints cannot be satisfied, then s_i is not E-admissible. We thus obtain the following algorithm, where LR is a list of constraints produced by pairs of strategies:

CRITERION–E-ADMISSIBILITY(Aux)
1   $N \leftarrow$ Number of strategies in Aux;
2   **for** $i = 1 \ldots N$ **do**
3      $LR \leftarrow null$;
4   **for** $j = 1 \ldots N$ **do**
5   **if** $i \neq j$ **then**
6            $LR \leftarrow LR \cup E[Aux[i] - Aux[j]] \geq 0$;
7   **endif**
8   **endfor**
9      $Q \leftarrow$ set of all constraints on probability values plus LR;
10   $P \leftarrow \arg\max_p E[s_i]$ s.t. constraints on Q;
11   **if** P is non-null **then**
12      $Aux[i].admissible =$ **true**;
13   **else**
14      $Aux[i].admissible =$ **false**;
15   **endif**
16 **endfor**
17 **return** All alternatives not marked as false;

Lines 3 to 8 generate all constraints that are required to satisfy E-admissibility of a strategy s_i, and line 9 collects constraints on probabilities. Line 10 requires the solution of a multilinear program. We emphasize that the whole algorithm depends on N, the number of strategies, and not directly on the number distributions in the credal sets. Even though the properties of the credal sets certainly affect the solution of the relevant multilinear programs, there is no need to represent the credal sets explicitly, or to enumerate their vertices — steps that are necessary in existing methods [Troffaes, 2004]. In a sense, the complexity of credal sets is "hidden" within the multilinear programs. This raises the question of how efficient can be multilinear programming; we mention that previous work has indicated that state-of-the-art multilinear programming methods can handle problems containing hundreds of variables [Campos and Cozman, 2004].

4 Examples

In this section we apply the various decision criteria to two problems where beliefs are represented by credal sets. We start by analyzing the example presented in Section 2, with a small change: instead of adopting precise probability values, we take $q \in [0.4, 0.5]$ and $p \in [0.25, 0.75]$.

Algorithm DECISIONTREE begins evaluation at D_3. The array Aux contains three strategies (3.1), $(3.2a)$ and $(3.2b)$. Function COMBINATION is called twice, producing expressions: $(3.2a) = p*1+(1-p)*0$ and $(3.2b) = (1-p)*1+p*0$. The choice node D_2 is analogous to D_3. At D_1, the function COMBINATION generates the following multilinear expressions for strategies $2.a$ and $2.b$: $E[2.a] = q * (p * 1 + (1-p) * 0) + (1-q) * (p * 0 + (1-p) * 1)$ and $E[2.b] = q*(p*0+(1-p)*1)+(1-q)*(p*1+(1-p)*0)$. Note that upper and lower expectations are obtained through

the maximization and minimization of these expressions subject to $0.25 \leq p \leq 0.75$ and $0.4 \leq q \leq 0.5$. At sequential option 3, the function COMBINATION combines all admissible strategies (those returned by CRITERION–X in the previous steps). The function CRITERION–X is called once more and finally, the function STRATEGIES returns the admissible strategies.

The evaluation by Γ-Maximin in our example yields strategy (2.a) or (2.b) or (3, (3.1,3.1)) with a payoff of 0.05 units each. This example shows that sometimes the Γ-Maximin criterion may display somewhat strange behavior: while choices (3.2a) and (3.2b) are inadmissible at D_2 and D_3, their combination is the same as (2a) and (2b) which are admissible at D_1 [Seidenfeld, 2004]. Using Γ-Maximax at D_2 and D_3 we have two admissible alternatives: (3.2a) and (3.2b).

Using interval dominance at decision nodes D_2 and D_3, we have that all strategies are admissible, thus, at decision node D_1 we have nine possible combinations for the fourth decision plus the three first strategies. At node D_1, we obtain that strategies (1) and (3,(3.1,3.1)) are dominated. From the remaining strategies, four of them have $E[s] \in [-0.04, 0.26]$ (combinations between the decision (3.1) and the options (3.2a) or (3.2b)), two have $E[s] = [-0.1, 0.4]$ (same choice at D_2 and D_3) and the last two have $E[s] \in [0.1, 0.2]$.

According to the maximality criterion, we also have nine possible combinations at D_1 for the fourth decision. The dominated strategies are (1), (2a), (2b) and (3, (3.1,3.1)). The other eight strategies are admissible.

Finally, if we use E-admissibility in the sequential option 3, the first of these (3.1) is E-inadmissible. Thus, at the initial node we have the first three alternatives plus the four possible combinations for the fourth option. Applying E-admissibility on these strategies we have the first three inadmissible. Two have $E[s] \in [-0.1, 0.4]$ (same choice at D_2 and D_3) and the other two have $E[s] \in [0.1, 0.2]$.

Consider now a second example, the classic oil wildcatter problem, but with probability intervals. The problem is as follows. An oil wildcatter must decide either to drill or not to drill. The cost of drilling is \$70,000. If the decision is to drill, the hole may be wet, dry or soak with a return of \$120,000, \$0, and \$270,000, respectively. At the cost of \$10,000, the oil wildcatter could decide to take seismic soundings of the geological structure at the site. The soundings will disclose whether the terrain has no structure (almost no hope for oil), closed structure (indication for much oil) or an open structure (indication for some oil). Table 1 shows conditional probabilities (as interval-valued probabilities); take that prior probabilities of the test on no structure, open structure and closed structure are interval-valued as $[0.181, 0.222]$, $[0.333, 0.363]$ and $[0.444, 0.454]$.

Table 1: Conditional probabilities for the oil wildcatter problem.

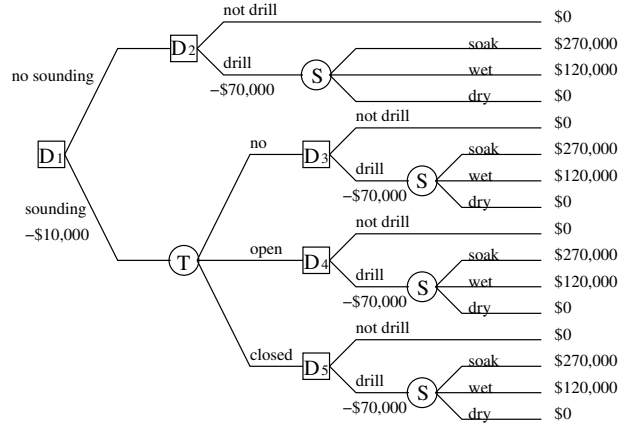| $S\|T$	dry	wet	soak
no	[0.500, 0.666]	[0.222, 0.272]	[0.125, 0.181]
open	[0.222, 0.333]	[0.363, 0.444]	[0.250, 0.363]
closed	[0.111, 0.166]	[0.333, 0.363]	[0.454, 0.625]

Figure 2 shows the decision tree for this problem.

Figure 2: Decision tree for the oil wildcatter problem.

We solve this problem using a criterion that produces a single strategy (Γ-maximin) and a criterion that produces several strategies (E-Admissibility). Using Γ-maximin we start by finding the upper and lower expectations at D_2, D_3, D_4, D_5, that is, $\max_S(\sum_{p \in S} p_i * \overline{E}_i)$ and $\min_S(\sum_{p \in S} p_i * \underline{E}_i)$ subject to $\sum_{p_i} = 1, p_i \geq 0$. The admissible strategies in these nodes are respectively: (drill), (not drill), (drill) and (drill). At decision node D_1 we have just two strategies (two multilinear programs to solve): $s_1 = \{ns, d\}$ and $s_2 = \{s, (d, n, d, d)\}$. Choosing s_1 we obtain the expectation $[20,000, 32,000]$ and, choosing s_2 the expectation is $[25,537.19, 42,626.26]$. Thus, according to Γ-maximin, the best option is take the strategy s_2. According to E-admissibility, the admissible strategies at D_2, D_3, D_4, D_5 are: (drill), (not drill) or (drill), (drill) and (drill). At D_1 the strategies are $s_1 = \{ns, d\}, s_2 = \{s, (d, n, d, d)\}$ and $s_3 = \{s, (d, d, d, d)\}$. All three are admissible, that is, these strategies can produce a maximal expectation for some probability p.

5 Conclusion

In this paper we have presented algorithms for strategy generation in decision trees associated with imprecise beliefs. As such decision trees represent partially ordered preferences, there are several criteria that can be used to generate strategies. The paper contributes in two ways:

1. It presents a multilinear programming framework for strategy generation. We emphasize that existing techniques for multilinear programming can handle problems with hundreds of variables [Campos and Cozman, 2004], thus guaranteeing that our algorithms can be applied to large problems.

2. It presents an algorithm for E-admissibility that depends essentially on the number of strategies to be compared, and not so much on the underlying credal set.

Given the diversity of criteria, one may wonder whether there is a "best criterion" in the field. The following comments may be relevant to this question. It seems that Γ-maximin is

appealing conceptually, and relatively simple from a computational point of view — but "extensive" Γ-maximin solutions can be incoherent in a sequential manner [Seidenfeld, 2004]. The Γ-maximax criterion seems too optimistic, even though it may be appropriate in some situations [Satia and Lave, 1973]. The other three criteria, interval dominance, maximality and E-admissibility, produce sets of strategies with increasing selectivity — that is, they are progressively more faithful to the partial order of preferences. In particular, E-admissibiliy does reveal the partial order of preferences in its sets of admissible strategies. Our results show that interval dominance is linear while maximality and E-admissibility are quadratic (here linear and quadratic are used informally to refer to the number and size of multilinear programs). This is a significantly simpler picture than previously believed [Troffaes, 2004]. Altogether, E-admissibility emerges as a conceptually elegant and computationally feasible criterion for decision trees with imprecise probabilities.

Acknowledgments

References

[Berger, 1985] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 1985.

[Campos and Cozman, 2004] C. P. de Campos and F. G. Cozman. Inference in credal networks using multilinear programming. *Proc. of the 2nd Starting AI Researcher Symp.*, pp. 50–61, Valencia, 2004. IOS Press.

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

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

[Gilboa and Schmeidler, 1989] I. Gilboa and D. Schmeidler. Maxmin expected utility with non-unique prior. *Journal of Mathematical Economics*, 18(2):141–153, 1989

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

[Jaffray, 1999] J.-Y. Jaffray. Rational decision making with imprecise probabilities. In *Proceedings of 1st Int. Symp. on Imprecise Probabilities and Their Applications*, Ghent, Belgium, 1999.

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

[Luce and Raiffa, 1957] R. Duncan Luce and H. Raiffa. *Games and Decisions*. Wiley, New York, 1957.

[Machina, 1989] M. J. Machina. Dynamic consistency and non-expected utility models of choice under uncertainty. *Journal of Economic Literature*, 27:1622–1688, 1989.

[Raiffa, 1968] H. Raiffa. *Decision Analysis: Introductory Lectures on Choices under Uncertainty*. Addison-Welsey, Massachusetts, 1968.

[Samuelson, 1948] P. Samuelson. Are counterfactual decisions relevant for dynamically consistent updating under nonexpected utility? *Econometrica*, 15:243–253, 1948.

[Satia and Lave, 1973] J. K. Satia and R. E. Lave, Jr. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21(3):728–740, 1973.

[Schervish *et al.*, 2003] M. J. Schervish, T. Seidenfeld, J. B. Kadane, and I. Levi. Extensions of expected utility theory and some limitations of pairwise comparisons. In *Proc. of 3rd Int. Symp. on Imprecise Probabilities and Their Applications*, pages 496–510, Lugano, Switzerland, 2003.

[Seidenfeld, 1988] T. Seidenfeld. Decision theory without independence or without ordering: What is the difference. *Economics and Philosophy*, 4:267–290, 1988.

[Seidenfeld, 1995] T. Seidenfeld, M. J. Schervish, and J. B. Kadane. A representation of partially ordered preferences. *Annals of Statistics*, 23(6):2168–2217, 1995.

[Seidenfeld, 2004] T. Seidenfeld. A contrast between two decision rules for use with (convex) sets of probabilities: Gamma-maximin versus E-admissibility. *Synthese*, 140(3):69–88, June 2004.

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

[Troffaes, 2004] M. Troffaes. Decision making with imprecise probabilities: A short review. *SIPTA Newsletter*, pages 4–7, Society for Imprecise Probability Theory and Applications, Manno, Switzerland, December 2004.

[Walley, 1991] P. Walley. *Statistical reasoning with imprecise probabilities*. Chapman and Hall, Londom, 1991.

[Wang and Boutilier, 2003] T. Wang and C. Boutilier. Incremental utility elicitation with the minimax regret decision criterion. *Proc. of Int. Joint Conf. on Artificial Intelligence*, pages 309–316, 2003.