

Consistency and Set Intersection

Yuanlin Zhang and Roland H.C. Yap

National University of Singapore

3 Science Drive 2, Singapore

{zhangyl, ryap}@comp.nus.edu.sg

Abstract

We propose a new framework to study properties of consistency in a Constraint Network from the perspective of properties of set intersection. Our framework comes with a proof schema which gives a generic way of lifting a set intersection property to one on consistency. Various well known results can be derived with this framework. More importantly, we use the framework to obtain a number of new results. We identify a new class of tree convex constraints where local consistency ensures global consistency. Another result is that in a network of arbitrary constraints, local consistency implies global consistency whenever there are certain m -tight constraints. The most interesting result is that when the constraint on every pair of variables is properly m -tight in an arbitrary network, global consistency can be achieved by enforcing relational $m+1$ -consistency. These results significantly improve our understanding of convex and tight constraints. This demonstrates that our framework is a promising and powerful tool for studying consistency.

1 Introduction

A fundamental problem in Constraint Networks is to study properties of consistency in some networks with particular constraints or structure. There have been two main approaches to this problem.

The first is to utilise topological structure of the network. For example, where the network forms a tree structure, arc consistency is sufficient to make the network globally consistent [Freuder, 1982].

Another approach is to make use of semantic properties of the constraints. For *monotone constraints*, path consistency implies global consistency [Montanari, 1974]. Dechter [1992] shows that a certain level of consistency in a network whose domains are of limited size ensures global consistency. Van Beek and Dechter [1995] generalize monotone constraints to a larger class of *row convex constraints*. Later, they [1997] study the consistency inside a network with tight and loose constraints.

In this paper, we present a new framework¹ which unifies well known results mainly along the second approach (including [van Beek and Dechter, 1995; 1997]). The power of this framework is that it allows the study of the relationship between local and global consistency from the perspective of *set intersection*.

For example, one property of set intersection is that if every pair (2) of tree convex sets intersect, the whole collection of these sets also intersect. The main point is that local information on intersection of every pair of sets gives global information on intersection of all sets. Intuitively, this can be related to getting global consistency from local consistency. Our framework enables us to lift the result on tree convex sets to the following result on consistency. For a binary network of tree convex constraints, $(2+1)$ -consistency (path consistency) implies global consistency.

Properties of the intersection of *tree convex* sets, *small* sets, *large* sets are presented in section 3. Section 4 develops the framework which consists of a lifting lemma together with a proof schema which gives a generic way of using the lifting lemma to get consistency results from properties of set intersection. One benefit of the framework lies in that it provides a modular way to greatly simplify the understanding and proof of results on consistency. This benefit is considerable as often the proof of many existing results is complex and “hard-wired”.

We demonstrate the power of the framework by showing three new consistency results as well as a number of well known results. The first is a generalization of global consistency of row convex constraints to a network of tree convex constraints which is presented in section 5. The second is a new result on global consistency on *weakly tight* networks in section 6. These networks only require certain constraints to be m -tight rather than all constraints as shown in [van Beek and Dechter, 1997]. The final result is presented in section 8. This is most interesting because we can make certain networks globally consistent by enforcing relational consistency, but this may not be achievable in previous work (for example [van Beek and Dechter, 1997]).

¹Our work can be related to the first approach in those cases where the network topology leads to some set intersection property.

2 Preliminaries

A *constraint network* $\mathcal{R}$ is defined as a set of variables $N = \{x_1, x_2, \dots, x_n\}$; a set of finite domains $D = \{D_1, D_2, \dots, D_n\}$ where domain D_i , for all $i \in 1..n$, is a set of values that variable x_i can take; and a set of constraints $C = \{c_{S_1}, c_{S_2}, \dots, c_{S_e}\}$ where S_i , for all $i \in 1..n$, is a subset of $\{x_1, x_2, \dots, x_n\}$ and each constraint c_{S_i} is a relation defined on domains of all variables in S_i . The *arity* of constraint c_{S_i} is the number of variables in S_i .

An instantiation of variables $Y = \{x_1, \dots, x_j\}$ is denoted by $\bar{a} = (a_1, \dots, a_j)$ where $a_i \in D_i$ for $i \in 1..j$. An *extension* of $\bar{a}$ to a variable $x \notin Y$ is denoted by $(\bar{a}, u)$ where $u \in D_x$. An instantiation of a set of variables Y is *consistent* if it satisfies all constraints in $\mathcal{R}$ which don't involve any variable outside Y .

A constraint network $\mathcal{R}$ is *k-consistent* if and only if for any consistent instantiation $\bar{a}$ of any distinct $k-1$ variables, and for any new variable x , there exists $u \in D_x$ such that $(\bar{a}, u)$ is a consistent instantiation of the k variables. $\mathcal{R}$ is *strongly k-consistent* if and only if it is j -consistent for all $j \leq k$. A strongly n -consistent network is called *globally consistent*.

3 Properties on Set Intersection

The set intersection property which we are concerned with is:

Given a collection of l finite sets, under what conditions is the intersection of all l sets not empty?

We use S to denote the collection of l sets: $\{E_1, E_2, \dots, E_l\}$, and U the union of all the sets in S , that is $U = \bigcup_{i \in 1..l} E_i$.

This property may not make sense for collections of arbitrary sets. Here, we study sets with two restrictions.

3.1 Sets with Convexity Restrictions

Definition 1 Given a set U and a tree $\mathcal{T}$ with vertices U . A set $A \subseteq U$ is *tree convex* iff there exists a subtree, with only vertices A , of $\mathcal{T}$. The sets in S are *tree convex* if there is a tree on U under which every set in S is tree convex.

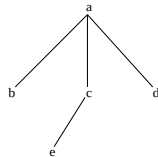


Figure 1: A tree with nodes a, b, c, d, e

Example. Consider a set $U = \{a, b, c, d, e\}$ and a tree in Fig 1. The subset $\{a, b, c, d\}$ is tree convex. So is the set $\{b, a, c, e\}$ since the elements in the set consists of a subtree. However, $\{b, c, e\}$ is not tree convex for they does not form a subtree of the given tree.

Consider $\mathcal{X} = \{\{1, 9\}, \{3, 9\}, \{5, 9\}\}$. A tree can be constructed on $\{1, 3, 5, 9\}$ with 9 being the root and 1, 3, 5 being its children. Each set in $\mathcal{X}$ covers the nodes of exactly one branch of the tree. Hence the sets in $\mathcal{X}$ are tree convex. $\square$

Tree convex sets have the following intersection property.

Lemma 1 (Tree convex Sets Intersection) Assume the sets in S are tree convex. $\bigcap_{i \in 1..l} E_i \neq \emptyset$ iff for every $E_i, E_j \in S$, $E_i \cap E_j \neq \emptyset$.

Proof. Let $\mathcal{T}$ be a tree such that there exists a subtree T_i for each $E_i \in S$. We can regard $\mathcal{T}$ as a rooted tree and thus every T_i ($i \in 1..l$) can be regarded as a rooted tree whose root is exactly the node nearest to the root of $\mathcal{T}$. Let r_i denote the root of T_i for $i \in 1..l$.

To prove $\bigcap_{i \in 1..l} E_i \neq \emptyset$, we want to show the intersection of the trees $\{T_i \mid i \in 1..l\}$ is not empty. The following proposition on subtrees, whose proof is omitted, is necessary in our main proof.

Proposition 1 Let the root of $\mathcal{T}$ be r . Given two subtrees T_1 and T_2 of $\mathcal{T}$ with roots r_1 and r_2 respectively. Let r_1 be not closer to r than r_2 , and T the intersection of T_1 and T_2 . r_1 is the root of T if T is not empty. Let $T = \bigcap_{i \in 1..l} T_i$. We are ready now to prove our main result

$T \neq \emptyset$. We select a tree $T_{\max}$ from $T_1, T_2, \dots, T_l$ such that its root $r_{\max}$ is the farthest away from r of $\mathcal{T}$ among the roots of the concerned trees. In terms of proposition 1, that $T_{\max}$ intersect with every other trees implies that $r_{\max}$ is a node of every T_i ($i \in 1..l$). Therefore, $r_{\max} \in T$. $\square$

Tree convexity on a collection of sets imposes a strong restriction on the relationship among these sets in the sense that each set forms part of a tree. If the tree is a chain, it can be thought of as a total order on the elements of all the sets. This leads to the notion of convex sets upon which the well-known row-convex constraints are defined.

Definition 2 Given a set U and a total ordering " $\preceq$ " on it, a set $A \subseteq U$ is *convex* if the elements in it are consecutive under the ordering, that is

$$A = \{v \in U \mid \min A \preceq v \preceq \max A\}.$$

Given S and U , the sets in S are *convex* if there is a total ordering on U such that every set in S is convex under the ordering.

Example. The set of real numbers between 1 and 2 is convex under the usual ordering of numbers. $\{1, 9\}$ and $\{3, 9\}$ are convex with a total ordering $1 \preceq 9 \preceq 3$. However, $\{1, 9\}$, $\{3, 9\}$ and $\{5, 9\}$ are not convex under any total ordering on $\{1, 3, 5, 9\}$. $\square$

The following result on convex sets, a set-based version of van Beek and Dechter's [1995, lemma 3.1], is immediate.

Corollary 1 (Convex Sets Intersection) Assume the sets in S are convex under a total ordering on U . $\bigcap_{i \in 1..l} E_i \neq \emptyset$ iff for any $E_i, E_j \in S$, $E_i \cap E_j \neq \emptyset$.

3.2 Sets with Cardinality Restrictions

The following results are on arbitrary sets where the only restriction is on the size of sets.

Lemma 2 (Small Set Intersection) Given a collection of sets S . Assume there is a set $E \in S$ such that $|E| \leq m$.

$$\bigcap_{i \in 1..l} E_i \neq \emptyset$$

iff the intersection of any $m + 1$ sets from S is not empty.

Proof. The necessary condition is immediate.

The sufficient condition is proven by induction on l . It is obviously true when $l \leq m + 1$. Assuming that $k (> m)$ sets intersect, we show that any $k + 1$ sets intersect. Without loss of generality, the subscripts of the $k + 1$ sets are numbered from 1 to $k + 1$ and let $|E_1| \leq m$. Let A_i be the intersection of all the $k + 1$ sets except E_i :

$A_i = E_1 \cap \dots \cap E_{i-1} \cap E_{i+1} \cap \dots \cap E_{k+1}$, for $1 < i \leq k + 1$.
If $A_i \cap A_j \neq \emptyset$ for some $i, j \in 2..k + 1, i \neq j$,

$$\bigcap_{i \in 1..k+1} E_i = A_i \cap A_j \neq \emptyset.$$

Assume the contrary that $A_i \cap A_j = \emptyset$ for all distinct i and j . According to the construction of A_i 's,

$$E_1 \supseteq \bigcup_{i \in 2..k+1} A_i,$$

and $|A_i| \geq 1$ by the induction assumption. Henceforth,

$$|E_1| \geq \sum_{i \in 2..k+1} |A_i| \geq k > m$$

which contradicts $|E_1| \leq m$. $\square$

We have an immediate corollary of the lemma when all sets of concern have at most m elements.

Corollary 2 (Small Sets Intersection) For any $E_i \in S$, assume E_i is finite and $|E_i| \leq m (< l)$. $\bigcap_{i \in 1..l} E_i \neq \emptyset$ iff the intersection of any $m + 1$ sets from S is not empty.

Another special case is that some set has only one element.

Corollary 3 (Singleton Set Intersection) Given a collection of sets S . Assume there is a set $E \in S$ such that $|E| = 1$. $\bigcap_{i \in 1..l} E_i \neq \emptyset$ iff all sets mutually intersect.

Motivated by [van Beek and Dechter, 1997, lemma 4.1 in page 561], we consider the intersection of sets, each of which has at least some number of elements.

Lemma 3 (Large Sets Intersection) For all $E_i \in S$, assume E_i is finite and $|E_i| \geq m$. Let $|\bigcup_{i \in 1..l} E_i| = d$. If $l \leq \lceil d/(d - m) \rceil - 1$, then $\bigcap_{i \in 1..l} E_i \neq \emptyset$.

Proof. Let $U = \bigcup_{i \in 1..l} E_i$, and $A_i = U - E_i$ for all $i \leq l$. It is immediate that

$$\bigcup_{i \in 1..l} A_i \subseteq U.$$

We know

$$|\bigcup_{i \in 1..l} A_i| \leq \sum_{i \in 1..l} |A_i|.$$

For $|A_i| \leq d - m$, we have

$$\sum_{i \in 1..l} |A_i| \leq \sum_{i \in 1..l} (d - m) = l(d - m) < d.$$

Hence, $\bigcup_{i \in 1..l} A_i$ is a proper subset of U . There exists $x \in U$ such that $x \notin A_i$ for all $i \leq l$, which implies that $x \in E_i$ for all $i \leq l$. $\square$

4 Set Intersection and Consistency

In this section we relate set intersection with consistency in constraint networks.

Underlying the concept of consistency is whether an instantiation of some variables can be extended to a new variable such that all relevant constraints to the new variable are satisfied. A *relevant* constraint to a variable x_i is a constraint where only x_i is uninstantiated and the others are instantiated. Each relevant constraint allows a set (possibly empty) of values for the new variable. This set is called *extension set* below. The satisfiability of all relevant constraints depends on whether the intersection of their extension sets is non-empty (see lemma 4).

Definition 3 Given a constraint c_{S_i} , a variable $x \in S_i$ and any instantiation $\bar{a}$ of $S_i - \{x\}$, the extension set of $\bar{a}$ to x with respect to c_{S_i} is defined as

$$E_{i,x}(\bar{a}) = \{b \in D_x \mid (\bar{a}, b) \text{ satisfies } c_{S_i}\}.$$

An extension set is trivial if it is empty; otherwise it is non-trivial.

Throughout this paper, it is often the case that an instantiation $\bar{a}$ of $S - \{x\}$ is already given where $S - \{x\}$ is a superset of $S_i - \{x\}$. Let $\bar{b}$ be the instantiation obtained by restricting $\bar{a}$ to the variables only in $S_i - \{x\}$. For ease of presentation, we continue to use $E_{i,x}(\bar{a})$, rather than $E_{i,x}(\bar{b})$, to denote the extension of $\bar{b}$ to x under constraint c_{S_i} . Where the context is clear, we may omit some of the three parameters $i, \bar{a}$ and x .

Example. Consider the network with variables $\{x, x_1, x_2, x_4, x_5\}$:

$$\begin{aligned} c_{S_1} &= \{(a, b, d), (a, b, a)\}, & S_1 &= \{x_1, x_2, x\}; \\ c_{S_2} &= \{(b, a, d), (b, a, b)\}, & S_2 &= \{x_2, x_4, x\}; \\ c_{S_3} &= \{(b, d), (b, c)\}, & S_3 &= \{x_2, x\}; \\ c_{S_4} &= \{(b, a, d), (b, a, a)\}, & S_4 &= \{x_2, x_5, x\}; \\ D_1 &= D_4 = D_5 = \{a\}, & D_2 &= \{b\}, D_x = \{a, b, c, d\}. \end{aligned}$$

Let $\bar{a} = (a, b, a)$ be an instantiation of variables $Y = \{x_1, x_2, x_4\}$. The relevant constraints to x are c_{S_1} , c_{S_2} , and c_{S_3} . c_{S_4} is not relevant since it has two uninstantiated variables. The extension sets of $\bar{a}$ to x with respect to the relevant constraints are:

$$E_1(\bar{a}) = \{d, a\}, E_2(\bar{a}) = \{d, b\}, E_3(\bar{a}) = \{d, c\}.$$

The intersection of the above extension sets is not empty, implying that $\bar{a}$ can be extended to satisfy all relevant constraints c_{S_1} , c_{S_2} and c_{S_3} .

Let $\bar{a} = (b, b)$ be an instantiation of $\{x_2, x\}$. $E_{1,x_1}(\bar{a}) = \emptyset$ and it is trivial. In other words, when an instantiation has a trivial extension set, it can not be extended to satisfy the constraint of concern. $\square$

The relationship between k -consistency and set intersection is characterized by the following lemma which is a direct consequence of the definition of k -consistency.

Lemma 4 (Set Intersection and Consistency; lifting) A constraint network $\mathcal{R}$ is k -consistent if and only if for any consistent instantiation $\bar{a}$ of any $(k - 1)$ distinct variables $Y = \{x_1, x_2, \dots, x_{k-1}\}$, and any new variable x_k ,

$$\bigcap_{j \in 1..l} E_{i_j} \neq \emptyset$$

where E_{i_j} is the extension set of $\bar{a}$ to x_k with respect to $c_{S_{i_j}}$, and $c_{S_{i_1}}, \dots, c_{S_{i_l}}$ are all relevant constraints.

The insight behind this lemma is to see consistency from the perspective of set intersection.

Example. Consider again the previous example. We would like to check whether the network is 4-consistent. Consider the instantiation $\bar{a}$ of Y again. This is a trivial consistent instantiation since the network doesn't have a constraint among the variables in Y . To extend it to x , we need to check the first three constraints. The extension is feasible because the intersection of E_1, E_2 , and E_3 is not empty. We show the network is 4-consistent, by exhausting all consistent instantiations of any three variables. Conversely, if we know the network is 4-consistent, we can immediately say that the intersection of the three extension sets of $\bar{a}$ to x is not empty. $\square$

The usefulness of this lemma is that it allows consistency information to be obtained from the intersection of extension sets, and vice versa. Using this view of consistency as set intersection, some results on set intersection properties, including all those in section 3, can be *lifted* to get various consistency results for a constraint network through the following proof schema.

Proof Schema

1. (*Consistency to Set*) From a certain level of consistency in the constraint network, we derive information on the intersection of the extension sets according to lemma 4.
2. (*Set to Set*) From the local intersection information of sets, information may be obtained on intersection of more sets.
3. (*Set to Consistency*) From the new information on intersection of extension sets, higher level of consistency is obtained according to lemma 4.
4. (*Formulate conclusion on the consistency of the constraint network*). $\square$

Given the proof schema, lemma 4 is also called the *lifting* lemma.

In the following sections, we demonstrate how the set intersection properties and the proof schema are used to obtain new and also well known results on consistency of a network.

5 Application I: Global Consistency of Tree Convex Constraints

The notion of *extension set* plays the role of a bridge between the restrictions to set(s) and properties of special constraints. The sets in lemma 1 are restricted to be tree convex. If all extension sets of a constraint is tree convex, we call this constraint *tree convex*.

Definition 4 A constraint c_S is tree convex with respect to x if and only if the sets in

$$A = \{E_{S,x} \mid E_{S,x} \text{ is a non-trivial extension of some instantiation of } S - \{x\}\}$$

are tree convex. c_S is tree convex if it is tree convex wrt every $x \in S$ under a common tree on the union of domains involved. $\{(1, 1), (1, 9), (3, 3), (3, 9), (5, 5), (5, 9), (9, 1), (9, 9)\}$ is an example of a tree convex constraint but it is not row convex.

Definition 5 A constraint network is tree convex if and only if all constraints are tree convex under a common tree on the union of all domains.

Convex sets naturally give rise to convex constraints which is a special case of tree convex constraints.

Definition 6 A constraint c_S is row convex with respect to x if and only if the sets in

$$A = \{E_{S,x} \mid E_{S,x} \text{ is a non-trivial extension of some instantiation of } S - \{x\}\}$$

are convex. It is row convex if under a total ordering on the union of involved domains, it is row convex wrt every $x \in S$. A constraint network is row convex iff all constraints are row convex under a common total ordering on the union of all domains.

The consistency results on these networks can be derived from the property of set intersection using the proof schema. We obtain the main result of this section.

Theorem 1 (tree convexity) Let $\mathcal{R}$ be a network of constraints with arity at most r and $\mathcal{R}$ be strongly $2(r - 1) + 1$ consistent. If $\mathcal{R}$ is tree convex then it is globally consistent.

Proof. The network is strongly $2(r - 1) + 1$ consistent by assumption. We prove by induction that the network is k consistent for any $k \in \{2r, \dots, n\}$.

Consider any instantiation $\bar{a}$ of any $k - 1$ variables and any new variable x . Let the number of relevant constraints be l . For each relevant constraint there is one extension set of $\bar{a}$ to x . So we have l extension sets. If the intersection of all l sets is not empty, we have a value for x such that the extended instantiation satisfies all relevant constraints.

(*Consistency to Set*) Consider any two of the l extension sets: E_1 and E_2 . The two corresponding constraints involve at most $2(r - 1) + 1$ variables since the arity of a constraint is at most r and each of the two constraints has x as a variable. According to the consistency lemma, that $\mathcal{R}$ is $2(r - 1) + 1$ -consistent implies that the intersection of E_1 and E_2 is not empty.

(*Set to Set*) Since all relevant constraints are tree convex under the given tree, the extension sets of $\bar{a}$ to x are tree convex. Henceforth, the fact that every two of the extension sets intersect shows that the intersection of all l extension sets is not empty, in terms of the *tree convex sets intersection* lemma.

(*Set to Consistency*) From the consistency lemma, we have that $\mathcal{R}$ is k -consistent. $\square$

Since a row convex constraint is tree convex, we have the following result.

Corollary 4 (row convexity) [van Beek and Dechter, 1995] Let $\mathcal{R}$ be a network of constraints with arity at most r and $\mathcal{R}$ be strongly $2(r - 1) + 1$ consistent. If there exists an ordering of the domains $D_1, \dots, D_n$ of $\mathcal{R}$ such that $\mathcal{R}$ is row convex, $\mathcal{R}$ is globally consistent.

This can also be proved directly by lifting corollary 1.

6 Application II: Global Consistency on Weakly Tight Networks

In this section, we study networks with some tight constraints. The m -tight property of a constraint is related to the cardinality of the extension set in the following way.

Definition 7 A constraint c_{S_i} is m -tight with respect to $x \in S_i$ iff for any instantiation $\bar{a}$ of $S_i - \{x\}$,

$$|E_{i,x}| \leq m \text{ or } |E_{i,x}| = |D_x|.$$

A constraint c_{S_i} is m -tight iff it is m -tight with respect to every $x \in S_i$.

Definition 8 A constraint network is weakly m -tight at level k iff for every set of variables $\{x_1, \dots, x_l\} (k \leq l \leq n)$ and a new variable, there exists an m -tight constraint in the relevant constraints after the instantiation of the l variables.

The small set intersection lemma gives the following theorem.

Theorem 2 (Weak Tightness) If a constraint network $\mathcal{R}$ with constraints of arity at most r is strongly $((m+1)(r-1)+1)$ -consistent and weakly m -tight at level $((m+1)(r-1)+1)$, it is globally consistent.

The proof is similar to the theorem on tree convexity.

Immediately we have the following result which is a main result in [van Beek and Dechter, 1997].

Corollary 5 (Tightness) [van Beek and Dechter, 1997] If a constraint network $\mathcal{R}$ with constraints that are m -tight and of arity at most r is strongly $((m+1)(r-1)+1)$ -consistent, then it is globally consistent.

This result can of course be directly lifted from corollary 2. Corollary 5 requires every constraint to be m -tight. The weak tightness theorem on the other hand does not require all constraints to be m -tight. The following example illustrates this difference.

Example. For a weakly m -tight network, we are interested in its topological structure. Thus we have omitted the domains of variables here. Consider a network with five variables labelled $\{1, 2, 3, 4, 5\}$. In this network, for any pair of variables and for any three variables, there is a constraint. Assume the network is already strongly 4-consistent.

Since the network is already strongly 4-consistent, we can simply ignore the instantiations with less than 4 variables. This is why we introduce the level at which the network is weakly m -tight. The interesting level here is 4. Table 1 shows the relevant constraints for each possible extension of four instantiated variables to the other one. In the first row, $1234 \rightarrow 5$ stands for extending the instantiation of variables $\{1, 2, 3, 4\}$ to variable 5. Entries in its second column denote a constraint. For example, 125 denotes c_{125} . If the constraints on $\{1, 2, 5\}$ and $\{1, 3, 4\}$ (suffixed by * in the table) are m -tight, the network is weakly m -tight at level 4. Alternatively, if the constraints $\{1, 5\}$, $\{2, 3\}$ and $\{3, 4\}$ (suffixed by +) are m -tight, the network will also be weakly m -tight. However, the tightness corollary requires all binary and ternary constraints to be m -tight. The weak m -tightness theorem needs significantly less constraints to be m -tight. Further results are in section 8. $\square$

extension	relevant constraints				
$1234 \rightarrow 5$	125*,	135 ,	145 ,	235,	245,
	345,	15+,	25 ,	35 ,	45
$2345 \rightarrow 1$	231 ,	241 ,	251*,	341,	351,
	451,	21 ,	31 ,	41 ,	51+
$3451 \rightarrow 2$	132 ,	142 ,	152*,	342,	352,
	452,	12 ,	32+,	42 ,	52
$4512 \rightarrow 3$	123 ,	143*,	153 ,	243,	253,
	453,	13 ,	23+,	43 ,	53
$5123 \rightarrow 4$	124 ,	134*,	154 ,	234,	254,
	354,	14 ,	24 ,	34+,	54

Table 1: Relevant constraints in extending the instantiation of four variables to the other one

7 Application III: Constraint Looseness

To make the presentation complete, we list the result on looseness of constraints by van Beek and Dechter [1997] (see also [Zhang and Yap, 2003]) as an instance of our framework.

The m -loose property of a constraint is related to the cardinality of extension set in the following way.

Definition 9 A constraint c_{S_i} is m -loose with respect to $x \in S_i$ iff for any instantiation $\bar{a}$ of $S_i - \{x\}$, $|E_{i,x}| \geq m$. A constraint c_{S_i} is m -loose iff it is m -loose with respect to every $x \in S_i$.

The large set intersection lemma is lifted to the following result on constraint looseness.

Theorem 3 (Looseness) Given a constraint network with domains that are of size at most d and constraints that are m -loose and of arity at least r , $r \geq 2$. It is strongly k -consistent, where k is the maximum value such that $\text{binomial}(k-1, r-1) \leq \lceil d/(d-m) \rceil - 1$.

8 Application IV: Making Weakly Tight Networks Globally Consistent

In this section, relational consistency will be used to make a constraint network globally consistent.

Definition 10 [van Beek and Dechter, 1997] A constraint network is relationally m -consistent iff given (1) any m distinct constraints $c_{S_1}, \dots, c_{S_m}$, and (2) any $x \in \cap_{i=1}^m S_i$, and (3) any consistent instantiation $\bar{a}$ of the variables in $(\cup_{i=1}^m S_i - \{x\})$, there exists an extension of $\bar{a}$ to x such that the extension is consistent with the m relations. A network is strongly relationally m -consistent if it is relationally j -consistent for every $j \leq m$.

All the results on *small set intersection* and *tree convex set intersection* in section 3 can be rephrased by using relational consistency. For example, a new version of weak tightness is shown below. Its proof can be easily obtained following the proof schema.

Theorem 4 (Weak Tightness) If a constraint network $\mathcal{R}$ of constraints with arity of at most r is strongly relationally $(m+1)$ -consistent and weakly m -tight at level of $(m+1)(r-1)+1$, it is globally consistent.

Consider the weak m -tightness theorem 2 in section 6 based on local k -consistency. A weakly m -tight network in general may not have the level of consistency required by the theorem. To obtain global consistency, one may try to enforce this level of consistency. However, this may result in new constraints with higher arity in the network. The new constraints may in turn require a higher level of consistency to ensure global consistency. Therefore it is difficult to predict an exact level of consistency to enforce on the network to make it globally consistent.

Using relational consistency, it is possible to obtain global consistency by enforcing local consistency on the network. In order to achieve our main result we need a stronger version of m -tightness — *proper m -tightness*.

Definition 11 A constraint c_{S_i} is properly m -tight with respect to $x \in S_i$ iff for any instantiation $\bar{a}$ of $S_i - \{x\}$,

$$|E_{i,x}| \leq m.$$

A constraint c_{S_i} is properly m -tight iff it is properly m -tight with respect to every $x \in S_i$.

A constraint is m -tight if it is properly m -tight. The converse may not be true.

A weakly properly m -tight network is defined by replacing “ m -tight” with “properly m -tight” in definition 8 (section 6).

We have the following observation on the weak m -tightness and weak proper m -tightness of a network.

Proposition 2 A constraint network is weakly m -tight and weakly properly m -tight respectively if the constraint between every two variables in the network is m -tight and properly m -tight respectively.

It is easy to verify that the *proper m -tightness* of the binary constraints is preserved during the procedure to enforce certain level of consistency in the network. So we have the main result of this section.

Theorem 5 (Weak Proper-Tightness) *Given a constraint network whose constraint on every two variables is properly m -tight. It is globally consistent after it is made relationally $m + 1$ -consistent.*

This theorem follows immediately from the discussion above and theorem 4. The implication of this theorem is that as long as we have certain properly m -tight constraints on certain combinations of variables, the network can be made globally consistent by enforcing relational $m + 1$ -consistency.

9 Summary

We demonstrate how to infer properties of consistency on a network purely by making use of set intersection properties. In addition to the results shown here, some other results can also be obtained by the lifting lemma. For example, the work of David [1993] can be obtained by lifting the singleton set corollary 3. The work of Faltings and Sam-Haroud [1996] is on convex constraint networks in continuous domains and the idea there is to lift Helly’s theorem on intersection of convex sets in Euclidean spaces.

We show a number of new consistency results which we believe are significant progress to convexity and tightness

of constraints since van Beek and Dechter’s work [1995; 1997]. We identify a class of tree convex constraints which is a superset of row convex constraints [van Beek and Dechter, 1995]. In a network of tree convex constraints, global consistency is ensured by a certain level of local consistency.

We show that *in a network of arbitrary constraints, local consistency implies global consistency whenever there are m -tight constraints on certain variables* (e.g. theorem 2). However, when the network does not have the required local consistency, global consistency may not be simply obtained by enforcing such level of local consistency. A surprising result is that as long as the constraint between every pair of variables is *properly m -tight* in an arbitrary network, global consistency can be achieved by enforcing a certain level of *relational consistency* (theorem 5). In previous work (e.g. [van Beek and Dechter, 1997]), all constraints are required to be m -tight which may be violated by newly introduced constraints in the process of enforcing the intended relational consistency.

A promising line of work is to find more properties under which a network is weakly properly m -tight. Another obvious direction is to find other classes of sets with intersection properties which will likely give useful consistency results.

References

- [David, 1993] Philippe David. When functional and bijective constraints make a csp polynomial. In *Proceedings of Thirteenth International Joint Conference on Artificial Intelligence*, volume 1, pages 224–229, Chambéry, France, 1993.
- [Dechter, 1992] R. Dechter. From local to global consistency. *Artificial Intelligence*, 34:1–38, 1992.
- [Freuder, 1982] E.C. Freuder. A sufficient condition for backtrack-free search. *Journal of The ACM*, 29(1):24–32, 1982.
- [Montanari, 1974] U. Montanari. Networks of constraints: Fundamental properties and applications. *Information Science*, 7(2):95–132, 1974.
- [Sam-Haroud and Faltings, 1996] D. Sam-Haroud and Boi V. Faltings. Solving non-binary convex csps in continuous domains. In *Proceedings of International Conference on Principles and Practice of Constraint Programming*, pages 410–424, Cambridge, Massachusetts, 1996.
- [van Beek and Dechter, 1995] P. van Beek and R. Dechter. On the minimality and global consistency of row-convex constraint networks. *Journal of The ACM*, 42(3):543–561, 1995.
- [van Beek and Dechter, 1997] P. van Beek and R. Dechter. Constraint tightness and looseness versus local and global consistency. *Journal of The ACM*, 44(4):549–566, 1997.
- [Zhang and Yap, 2003] Yuanlin Zhang and Roland H. C. Yap. Erratum: P. van Beek and R. Dechter’s theorem on constraint looseness and local consistency. To appear, 2003.

CONSTRAINTS

CONSTRAINTS AND SYMMETRY

Efficient Symmetry Breaking for Boolean Satisfiability

Fadi A. Aloul

Karem A. Sakallah

Igor L. Markov

Department of Electrical Engineering and Computer Science

University of Michigan

Ann Arbor, MI 48109-2122

{faloul, karem, imarkov}@eecs.umich.edu

Abstract

Identifying and breaking the symmetries of CNF formulae has been shown to lead to significant reductions in search times. In this paper we describe a more systematic and efficient construction of symmetry-breaking predicates (SBPs). In particular, we use the cycle structure of symmetry generators, which typically involve very few variables, to drastically reduce the size of SBPs. Furthermore, our new SBP construction grows linearly with the number of relevant variables as opposed to the previous quadratic constructions. Our empirical data suggest that these improvements reduce search run times by one to two orders of magnitude on a wide variety of benchmarks with symmetries.

1 Introduction

Modern Boolean satisfiability (SAT) solvers, based on back-track search, are now capable of attacking instances with thousands of variables and millions of clauses [Velev and Bryant, 2001] and are being routinely deployed in a wide range of industrial applications [Biere *et al.*, 1999], [Nam *et al.*, 2001], [Stephan *et al.*, 1996]. Their success can be credited to a combination of recent algorithmic advances and carefully-tuned implementations [Silva and Sakallah, 1999], [Moskewicz *et al.*, 2001], [Goldberg and Novikov, 2002]. Still, there are problem instances that remain beyond the reach of most SAT solvers.

One aspect of intractability is the presence of symmetry in the conjunctive normal form (CNF) of a SAT instance. Intuitively, a symmetry of a discrete object is a transformation, e.g., a permutation, of its components that leaves the object intact. The symmetries of a CNF formula are permutations of its literals (variables and their negations) that result in a re-ordering of its clauses (and the literals within clauses) without changing the formula itself. Such symmetries induce an equivalence relation on the set of variable assignments such that two assignments are equivalent if and only if the formula assumes the same truth value (either 0 or 1) at each of these assignments. A search algorithm that is oblivious to the existence of these symmetries may end up, wastefully exploring a set of equivalent unsatisfying assignments before moving on to a more promising region of the search space.

On the other hand, knowledge of the symmetries can be used to significantly prune the search space. Symmetries are studied in abstract algebra in terms of *groups*. We assume the reader to be familiar with the basics of group theory; in particular, we assume familiarity with permutation groups and their representation in terms of irredundant sets of generators. A good reference on the subject is [Fraleigh, 2000].

The rest of the paper is organized in five sections. Section 2 provides a brief review of permutations and permutation groups. Section 3 describes previous work on symmetry breaking for SAT. Our main contribution on efficient constructions of symmetry-breaking predicates is detailed in Section 4. These constructions are evaluated empirically in Section 5, and we end with conclusions in Section 6.

2 Notation and Preliminaries

We will be concerned with permutations on the literals of a set of n Boolean variables $\{x_1, \dots, x_n\}$ which we assume to be totally ordered according to $x_1 < x_2 < \dots < x_n$. We use I_n to denote the set of integers between 1 and n inclusive, and denote non-empty subsets of I_n by upper-case “index variables” I and J as appropriate. Given an index set I and an index $i \in I$, we define the “index selector” functions:

$$\begin{aligned} \text{pred}(i, I) &= \{j \in I \mid j < i\} & \text{prev}(i, I) &= \max(\text{pred}(i, I)) \\ \text{succ}(i, I) &= \{j \in I \mid j > i\} & \text{next}(i, I) &= \min(\text{succ}(i, I)) \end{aligned} \quad (1)$$

where $\min$ and $\max$ return, respectively, the least and greatest element in the given index set. For completeness, we also let $\min(\emptyset) = n + 1$, and $\max(\emptyset) = 0$.

A permutation π of the set of $2n$ literals $L = \{x_1, x'_1, \dots, x_n, x'_n\}$ (where x'_i denotes the logical negation of x_i) is a function $\pi: L \rightarrow L$ that is both one-to-one and onto. We will denote that x_j is the image of x_i under π by writing $x_j = x_i^\pi$. To preserve Boolean consistency, whenever π maps x_i to x_j it must simultaneously map x'_i to x'_j . Such implied mappings will be assumed whenever not explicitly specified. A permutation π is a *phase-shift* permutation if $x_i^\pi = x'_i$ for some $i \in I_n$, i.e., π maps some literal to its complement.

Permutations will be expressed either in tabular form or in cyclic notation. For example,

$$\pi = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ x_1^\pi & x_2^\pi & \cdots & x_n^\pi \end{pmatrix} \quad (2)$$

denotes a permutation that maps x_1 to x_1^π etc. The same permutation can be expressed as a set of disjoint cycles such as

$$\pi = (x_i, x_i^\pi, (x_i^\pi)^\pi, \dots)(x_j, x_j^\pi, (x_j^\pi)^\pi, \dots) \dots \quad (3)$$

Here a cycle $(a, b, \dots, z)$ is a shortcut for “ a maps to b , b maps to c , $\dots$ and z maps to a .” The length of a cycle is equal to the number of literals in it; we will refer to a cycle whose length is k as a k -cycle. We define the support of a permutation π , $\text{supp}(\pi)$, to be the set of indices appearing in its cyclic representation, i.e.,

$$\text{supp}(\pi) = \{i \in I_n | x_i^\pi \neq x_i\} \quad (4)$$

The number of cycles in a permutation π will be denoted by $\text{cycles}(\pi)$. We also define $\text{phase-shift}(\pi)$ to be the index of the *smallest* variable (according to the assumed total ordering) that is mapped to its complement by π :

$$\text{phase-shift}(\pi) = \min \{i \in I_n | x_i^\pi = x'_i\} \quad (5)$$

We should note that a phase-shift permutation must have one or more *phase-shift cycles*, i.e., length-2 cycles that have the form (x_i, x'_i) . Finally, we define $\text{ends}(\pi)$ as follows:

$$\text{ends}(\pi) = \{i \in I_n | i \text{ is largest index of a variable in a non-phase-shift cycle of } \pi\} \quad (6)$$

A *permutation group* G is a group whose elements are permutations of some finite set and whose binary operation is function composition, also referred to as permutation multiplication. The order of a group is the number of its elements. A *subgroup* H of a group G , denoted $H \leq G$ is a subset of G that is closed under the group's binary operation. The *cyclic subgroup* of $\pi \in G$, denoted by $\langle \pi \rangle$, is the subgroup consisting of π and its integer powers:

$$\langle \pi \rangle = \{\pi^i | i \in \mathbb{Z}\} \quad (7)$$

and π is said to generate $\langle \pi \rangle$. A set of permutations $\pi_1 \in G, \dots, \pi_k \in G$ *generates* G if the subgroup resulting from taking all possible products of the integer powers of these permutations is equal to G . The permutations $\{\pi_1, \dots, \pi_k\}$ are called *generators* of G . A set of generators is *irredundant* if it is not possible to express any of its permutations as a product of powers of its other permutations. A set of irredundant generators serves as an implicit representation of the group it generates and, in general, guarantees exponential compression in the size of the representation. Note that a set of irredundant generators is not a group since it is not closed under multiplication and taking inverse. In the sequel, a set of permutations G that is not necessarily closed will be indicated by placing a “hat” on the variable denoting the set, i.e., $\hat{G}$. Additionally, and with a slight abuse of notation, we will indicate that G is the group generated by $\hat{G}$ by writing $G = \langle \hat{G} \rangle$.

3 Previous Work

The basic framework for utilizing the symmetries in a CNF instance to prune the search space explored by a SAT solver

was laid out in [Crawford *et al.*, 1996]. This framework was extended later, in [Aloul *et al.*, 2002], to account for phase-shift symmetries, take advantage of the cycle structure of permutations, and consider only generators of the group of symmetries. In outline, the procedure consists of the following steps:

1. Convert a CNF formula φ to a colored graph whose symmetries are isomorphic to the symmetries of the formula.
2. Find the symmetries of the graph in terms of a set of irredundant generators $\hat{G} = \{\pi_1, \dots, \pi_k\}$ using a suitable graph automorphism program [McKay, 1981], [Spitznagel, 1994].
3. Map the graph symmetries back to symmetries of the formula.
4. Construct an appropriate *symmetry-breaking predicate* (SBP) ρ and conjoin it to the formula.
5. Solve $\varphi \wedge \rho$ using a suitable SAT solver [Moskewicz *et al.*, 2001].

Our concern in this paper is step 4. Noting that the group of symmetries induces an equivalence relation on the set of assignments in the n -dimensional Boolean space, the basic idea is to construct a “filter” that picks out a single representative from each equivalence class. In particular, choosing the *lexicographically smallest representative*—according to the assumed total ordering on the variables—leads to the following *Lex-Leader* SBP [Crawford *et al.*, 1996]:

$$\rho^{\text{LL}}(\langle \hat{G} \rangle) = \bigwedge_{\pi \in \langle \hat{G} \rangle} \text{PP}(\pi) \quad (8)$$

$$\text{PP}(\pi) = \bigwedge_{i \in I} \text{BP}(\pi, i) \quad (9)$$

$$\text{BP}(\pi, i) = \left[\bigwedge_{j \in \text{pred}(i, I)} (x_j = x_j^\pi) \right] \rightarrow (x_i \leq x_i^\pi) \quad (10)$$

where the index set I in (9) and (10) is equal to I_n . In these equations, the Lex-Leader SBP is expressed as a conjunction of *permutation predicates* (PPs) each of which is a conjunction of *bit predicates* (BPs)¹. Introducing n auxiliary “equality” variables $e_i \equiv (x_i = x_i^\pi)$ makes it possible to express the i th BP as an $(i+1)$ -literal CNF clause. This leads to a CNF representation of the PP in (9) that has n clauses with a total literal count of $0.5(n^2 + 3n)$. Additionally, each of the introduced equality constraints yields 4 3-literal clauses bringing the total CNF size of (9) to:

$$\begin{aligned} \text{clauses}(\text{PP}(\pi)) &= 5n \\ \text{literals}(\text{PP}(\pi)) &= 0.5(n^2 + 27n) \end{aligned} \quad (11)$$

In its present form, the lex-leader SBP in (8)-(10) can lead to an exponentially large CNF formula because the order of the symmetry group can be exponential in the number of variables. Thus, its value in pruning the search space is negated by the need of the SAT solver to process a much larger CNF formula. To remedy this problem [Crawford *et al.*, 1996] suggested the construction of a symmetry tree to eliminate some redundant permutations. However, in the worst case the number of symmetries in the tree remains exponential. Empirical evidence in [Aloul *et al.*, 2002]

¹ Note that $x \leq y$ in the bit predicate means “ x implies y ”.

$$\pi = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \\ x_4 & x_2 & x_8 & x_1 & x'_5 & x'_3 & x_7 & x'_6 & x_9 & x_{10} \end{pmatrix}$$

$$= (x_1, x_4)(x_3, x_8, x'_6)(x_5, x'_5)$$

(a) Permutation in tabular and cyclic notation

$$\text{BP}(\pi, 1) = x_1 \leq x_4$$

$$\text{BP}(\pi, 2) = (x_1 = x_4) \rightarrow (x_2 \leq x_2)$$

$$\text{BP}(\pi, 3) = (x_1 = x_4)(x_2 = x_2) \rightarrow (x_3 \leq x_8)$$

$$\text{BP}(\pi, 4) = (x_1 = x_4)(x_2 = x_2)(x_3 = x_8) \rightarrow (x_4 \leq x_1)$$

$$\text{BP}(\pi, 5) = (x_1 = x_4)(x_2 = x_2)(x_3 = x_8)(x_4 = x_1) \rightarrow (x_5 \leq x'_5)$$

$$\text{BP}(\pi, 6) = (x_1 = x_4)(x_2 = x_2)(x_3 = x_8)(x_4 = x_1)(x_5 = x'_5) \rightarrow (x_6 \leq x'_3)$$

$$\text{BP}(\pi, 7) = (x_1 = x_4)(x_2 = x_2)(x_3 = x_8)(x_4 = x_1)(x_5 = x'_5)(x_6 = x'_3) \rightarrow (x_7 \leq x_7)$$

$$\text{BP}(\pi, 8) = (x_1 = x_4)(x_2 = x_2)(x_3 = x_8)(x_4 = x_1)(x_5 = x'_5)(x_6 = x'_3)(x_7 = x_7) \rightarrow (x_8 \leq x'_6)$$

$$\text{BP}(\pi, 9) = (x_1 = x_4)(x_2 = x_2)(x_3 = x_8)(x_4 = x_1)(x_5 = x'_5)(x_6 = x'_3)(x_7 = x_7)(x_8 = x'_6) \rightarrow (x_9 \leq x_9)$$

$$\text{BP}(\pi, 10) = (x_1 = x_4)(x_2 = x_2)(x_3 = x_8)(x_4 = x_1)(x_5 = x'_5)(x_6 = x'_3)(x_7 = x_7)(x_8 = x'_6)(x_9 = x_9) \rightarrow (x_{10} \leq x_{10})$$

(c) Bit predicates according to (10). BPs enclosed in boxes with square corners are tautologous because π maps the corresponding bits to themselves. BPs enclosed in boxes with rounded corners are tautologous because they correspond to cycle “ends.” The BPs for bits 6 to 10 are tautologous because π maps bit 5 to its complement.

$$\text{supp}(\pi) = \{1, 3, 4, 5, 6, 8\}$$

$$\text{phase-shift}(\pi) = 5$$

$$\text{succ}(\text{phase-shift}(\pi), I_{10}) = \{6, 7, 8, 9, 10\}$$

$$\text{ends}(\pi) = \{4, 8\}$$

$$\text{supp}(\pi) \setminus \text{ends}(\pi) = \{1, 3, 5, 6\}$$

$$\text{supp}(\pi) \setminus \text{ends}(\pi) \setminus \text{succ}(\text{phase-shift}(\pi), I_{10}) = \{1, 3, 5\}$$

(b) Various index sets associated with permutation

$$\text{PP}(\pi) = (p_1)(p_1 \rightarrow l_1 p_3)(p_3 \rightarrow g_1 \rightarrow l_3 p_5)(p_5 \rightarrow g_3 \rightarrow l_5)$$

$$= (p_1)(p_1 \rightarrow (x_1 \leq x_4) p_3)(p_3 \rightarrow (x_1 \geq x_4) \rightarrow (x_3 \leq x_8) p_5)(p_5 \rightarrow (x_3 \geq x_8) \rightarrow x'_5)$$

(d) Linear formulation of the permutation predicate according to (18), based only on irredundant bits

Figure 1: Illustration of different formulations of the permutation predicate.

showed that full symmetry breaking, i.e., insuring that the SBP selects only the lex-leader from each equivalence class, is not necessary to obtain significant pruning of the search space. An SBP that breaks some, but not necessarily all, the symmetries of the formula can, in fact, provide a much better space/time trade-off during the search. This is accomplished by replacing the group of symmetries in (8) by a suitable, and much smaller, set of permutations $\hat{H} \subseteq \hat{G}$:

$$\rho^{\text{LL}}(\hat{H}) = \bigwedge_{\pi \in \hat{H}} \text{PP}(\pi) \quad (12)$$

In particular, [Aloul *et al.*, 2002] advocated the use of the set of generators $\hat{G}$ returned by the graph automorphism program in step 2.

4 Efficient Formulation of Permutation Predicate

Even when only a small number of permutations is used in constructing an SBP as in (12), the corresponding CNF formula may still be too large because each PP requires a CNF formula whose size is quadratic in the number of variables n . In this section we introduce two refinements that lead to much smaller PPs. The first refinement utilizes the *cycle structure* of a permutation to eliminate redundant bit predicates and can be viewed as replacing n in (11) by a much smaller number m and represents a more comprehensive and systematic treatment of cycles than that in [Aloul *et al.*, 2002]. The second refinement takes advantage of the *recursive bit-by-bit structure* in (10) to yield a CNF formula

whose size is linear, rather than quadratic, in m . Figure 1 provides an example illustrating these refinements.

Elimination of redundant BPs. Careful analysis of (10) reveals three cases in which a BP is tautologous, and hence redundant. The first corresponds to bits that are mapped to themselves by the permutation, i.e., $x_i^\pi = x_i$. This makes the consequent of the implication in (10), and hence the whole bit predicate, unconditionally true. Removal of such BPs is easily accomplished by setting the index set I in (9) and (10) to $\text{supp}(\pi)$ rather than I_n . For sparse permutations, i.e., permutations for which $|\text{supp}(\pi)| \ll n$, this change alone can account for most of the reduction in the CNF size of the PP.

The second case corresponds to the BP of the last bit in each cycle of π . “Last” here refers to the assumed total ordering on the variables. Assume a cycle involving the variables $\{x_j | j \in J\}$ for some index set J and let $i = \max(J)$. Then

$$\left[\bigwedge_{j \in J \setminus \{i\}} (x_j = x_j^\pi) \right] \rightarrow (x_i \leq x_i^\pi) = 1 \quad (13)$$

causing the corresponding bit predicate $\text{BP}(\pi, i)$ to be tautologous. Elimination of these BPs is accomplished by restricting the index set I in (9) and (10) further to just $\text{supp}(\pi) \setminus \text{ends}(\pi)$ and corresponds to a reduction in the number of BPs from n to $m \equiv |\text{supp}(\pi)| - \text{cycles}(\pi)$.

The third and last case corresponds to the BPs of those bits that occur *after* the first “phase-shifted variable.” Let i be the index of the first variable for which $x_i^\pi = x'_i$. Thus,

$e_i = 0$ and all BPs for $j > i$ have the form $0 \rightarrow (x_j \leq x_j^\pi)$ making them unconditionally true.

Taken together, the redundant BPs corresponding to these three cases can be easily eliminated by setting the index set in (9) and (10) to:

$$I = \text{supp}(\pi) \setminus \text{ends}(\pi) \setminus \text{succ}(\text{phase-shift}(\pi), I_n) \quad (14)$$

In the sequel we will refer to the bits in the above index set as “irredundant bits.” Note that the presence of a phase-shifted variable early in the total order can lead to a drastic reduction in the number of irredundant bits. For example, if $\pi = (x_1, x'_1) \dots$ then $\text{PP}(\pi)$ is simply (x'_1) regardless of how many other variables are moved by π .

Linear construction of PPs through chaining. The PP in (9) and (10) has a recursive structure that can be utilized to produce a CNF formula whose size is linear, rather than quadratic, in the cardinality of the index set I . Specifically, we introduce the “ordering” predicates $l_i = (x_i \leq x_i^\pi)$ and $g_i = (x_i \geq x_i^\pi)$ and, after algebraic manipulation, write the following equivalent expressions for the permutation predicate:

$$\begin{aligned} \text{PP}(\pi) &= g_0 \rightarrow \bigwedge_{i \in I} \left\{ \left(\bigwedge_{j \in \text{pred}(i, I)} g_j \right) \rightarrow l_i \right\} \\ &= g_0 \rightarrow l_k \wedge \left[g_k \rightarrow \bigwedge_{i \in K} \left\{ \left(\bigwedge_{j \in \text{pred}(i, K)} g_j \right) \rightarrow l_i \right\} \right] \end{aligned} \quad (15)$$

where $g_0 \equiv 1$, $k = \text{next}(0, I)$ and $K = \text{succ}(k, I)$. Noting that, except for the index set used, the parenthesized expression on the second line of the above equation is identical to the expression on the first line, we introduce a sequence of *chaining predicates* $\{p_i | i \in I\}$ defined according to:

$$p_i = g_{\text{prev}(i, I)} \rightarrow \bigwedge_{k \in K} \left[\bigwedge_{j \in \text{pred}(k, K)} g_j \right] \rightarrow l_k \quad (16)$$

where $K = \{i\} \cup \text{succ}(i, I) = \{k \in I | k \geq i\}$. The recursive structure of (15), now, makes it possible to express each chaining predicate in terms of the one that follows it

$$p_i = g_{\text{prev}(i, I)} \rightarrow l_i p_{\text{next}(i, I)} \quad i \in I, p_{n+1} \equiv 1 \quad (17)$$

and yields the following alternative representation of the permutation predicate:

$$\text{PP}(\pi) = p_{\min(I)} \wedge \bigwedge_{i \in I} \left[p_i = g_{\text{prev}(i, I)} \rightarrow l_i p_{\text{next}(i, I)} \right]$$

which can be simplified further by replacing the equalities by one-way implications leading, finally, to:

$$\text{PP}(\pi) = p_{\min(I)} \wedge \bigwedge_{i \in I} \left[p_i \rightarrow g_{\text{prev}(i, I)} \rightarrow l_i p_{\text{next}(i, I)} \right] \quad (18)$$

The CNF representation of each conjunct in (18) is obtained by substituting the definitions of the l and g variables and using the distributive law. Thus, using this construction the permutation predicate requires $|I|$ additional variables (the chaining predicates) and consists of $2|I|$ 3-literal and $2|I|$ 4-literal clauses for a total of $14|I|$ literals.

5 Experimental Evaluation

We conducted a number of experiments to evaluate the effectiveness of the symmetry breaking constructions described above in reducing search times. We ran the experiments on representative CNF instances from the following seven benchmark families:

1. **Hole-n**: Unsatisfiable pigeon-hole instances [DIMACS]
2. **Urq**: Unsatisfiable randomized instances based on expander graphs [Urquhart, 1987]
3. **GRoute**: Difficult satisfiable instances that model global wire routing in integrated circuits [Aloul *et al.*, 2002]
4. **FPGARoute** and **ChnlRoute**: Large satisfiable and unsatisfiable, instances that model the routing of wires in the channels of field-programmable integrated circuits [Nam *et al.*, 2001]
5. **XOR**: Various exclusive-or chains [SAT 2002]
6. **2pipe**: Difficult unsatisfiable instances that model the functional correctness requirements of modern out-of-order microprocessor CPUs [Velev and Bryant, 2001]

Each of the benchmarks was converted to a colored graph as described in [Aloul *et al.*, 2002] and processed by the graph automorphism program Nauty (version 2) [McKay, 1981] using the GAP package (version 4, release 3) [Spitznagel, 1994]. The symmetries returned by Nauty were then mapped back to symmetries of the benchmark and appropriate SBPs constructed and added. The mChaff SAT solver [Moskewicz *et al.*, 2001] was then run on the original and SBP-augmented versions of each benchmark. All experiments were run on a Linux workstation with a 1.2Ghz AMD Athlon processor and 1GB of RAM. A time-out limit of 1000 seconds was set for all runs, and run time results represent averages over 10 to 200 independent runs.

Table 1 lists, for each benchmark family, the number of instances tested (col. 2), their total CNF sizes (cols. 3 and 4), the order of their symmetry groups (col. 5), the total number of generators returned by Nauty (col. 6), and the number of those that include phase shifts (col. 7). Columns 8 and 9 list the cardinality of the generators’ support and the degree of sparsity present in these generators. The remaining columns in the table list the CNF sizes of three SBP constructions based on generators:

- The quadratic construction (using extra equality variables) based on all bits; this represents the previous state-of-the-art
- The quadratic construction based only on irredundant bits
- The linear construction (using extra chaining variables) based only on irredundant bits

Several observations can be made about the data in Table 1. The number of symmetries in these benchmarks is large, but all symmetries, including phase shifts in benchmark families Urq, XOR and 2pipe, can be represented by fairly small sets of generators. The generators returned by Nauty appear very sparse on average, i.e., a typical generator affects only a small number of variables.² This explains the reduction, by 1-2 orders-of-magnitude, in the size of symmetry-breaking predicates in column 13 (our first construction) versus column 11 ([Crawford *et al.*, 1996]). Both the number of variables and the number of literals are reduced. While our construction in column 13 only slightly extends the qua-

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Bench- mark Family	# Instances	Total Instance Sizes		Symmetries					Total SBP Sizes					
				Total	Generators				All bits		Irredundant bits			
					Quadratic Const.						Linear Const.			
		Vars	Lits		Tot	PS		Avg supp	S	Extra Vars	Lits	Extra Vars	Lits	Extra Vars
Hole-n	6	616	6818	3E+18	108	0	20.6	79%	11788	865018	1112	20903	1112	14380
Urq	4	162	7212	6E+08	94	94	1.0	98%	3876	133047	0	94	0	94
GRoute	5	4704	93262	2E+11	138	0	46.6	95%	130032	63307944	3216	87312	3216	43506
FPGARoute	9	1580	19612	1E+21	241	0	24.2	86%	43429	4657249	2922	66603	2922	38257
ChnlRoute	6	1710	24366	2E+52	270	0	29.5	89%	79782	13705131	3987	138079	3987	52848
XOR	4	316	2512	7E+10	108	107	1.0	99%	9240	544092	1	121	1	110
2pipe	3	2620	61566	168	15	10	12.3	82%	13154	5954599	1031	186708	1031	14329
Total	37	11708	215348	2E+52	974	211	24749	90%	291301	89167080	12269	499820	12269	47751

KEY: Vars: # variables; Lits: # literals; Extra Vars: # additional variables used in SBPs; Tot: total # generators; PS: # phase-shift generators; Avg. |supp|: average support of generators; S: sparsity of generators (1-Avg |supp|/Vars)

Table 1: Symmetry statistics for various benchmark families and size comparisons of three SBP constructions based on group generators.

1	2	3	4	5	6	7	8	9	10	11	12	
Bench- mark Family	# Instances	# Generators	# Generators & their compositions	Time to find symmetries (sec)	Time to solve original instances (sec)	Time to solve instances and SBPs (sec)				Speedup		
						Generators only				Generators & their comp.	$\frac{\text{col 6}}{\text{col 5} + \text{col 9}}$	$\frac{\text{col 7}}{\text{col 5} + \text{col 9}}$
						All bits		Irredundant bits				
						Quadratic Construction		Linear Construction				
Hole-n	6	108	1157	0.38	1246.81	18.75	0.06	0.07	0.65	8832	171	
Urq	4	94	94	0.76	895.65	3.72	1.17	1.17	1.17	2084	11	
GRoute	5	138	2037	38.76	132.30	960.10	15.55	5.08	38.32	15	99	
FPGARoute	9	241	3595	3.24	5070.30	4832.70	0.31	0.21	4.08	13391	12835	
ChnlRoute	6	270	6578	25.86	2755.00	2675.85	0.37	0.17	5.87	588	569	
XOR	4	108	108	11.43	1900.16	20.52	2.41	2.41	2.41	531	5	
2pipe	3	15	18	23.50	7.84	16.88	3.19	8.01	8.31	0.8	2	
Total	37	974	13587	103.93	12008.06	8528.52	23.06	17.12	60.81	3635	1956	

Table 2: Comparison of search run times for various choices of SBP constructions and symmetries to break.

dratic-size construction in [Aloul *et al.*, 2002], our more advanced linear-size construction (column 15) offers an additional reduction by up to an order of magnitude. Note, however, that the number of variables is unchanged—the extra variables added by the two constructions have different function, but can be mapped to each other one-to-one.

Table 2 empirically compares the effectiveness of the symmetry-breaking predicates described in Table 1. First, in

² While our experiments clearly confirm that such generators exist, finding them is a function of the graph automorphism program. Since we used Nauty as a black box, we do not have a clear explanation of what causes Nauty to find such sparse generators, whether even sparser generators can be found and whether Nauty will in some cases return less sparse generators.

most cases it takes much less time to find symmetries of a CNF instance than to solve it. The 2pipe instances are an exception, but we believe that further advances in symmetry-finding can rectify this exception. Second, the all-bits quadratic construction due to [Crawford *et al.*, 1996] is dramatically slower than our variants based on the cycle structure. Our linear-sized construction provides a further speed-up. Exceptions are the Hole-n benchmarks, where the difference between our two constructions is small, and the 2pipe benchmarks where the quadratic-sized SBPs led to shorter run times despite being almost thirteen times larger. Further investigation showed that this was a side-effect of mChaff’s VSIDS decision heuristic which tended to choose the auxiliary variables first because they occur in many more clauses than the original instance variables. Using BerkMin [Gold-

Instance	2-cycle generators		Generators with long cycles						
	Tot	Time (sec)	Max cycle length	Generators only		Generators and their powers		Generators and their comp.	
				Tot	Time (sec)	Tot	Time (sec)	Tot	Time (sec)
Hole-7	13	0.01	21	12	0.01	94	0.08	52	0.07
Hole-8	15	0.01	12	14	0.03	127	0.05	24	0.17
Hole-9	17	0.01	63	16	0.55	161	2.95	77	0.97
Hole-10	19	0.01	42	18	8.2	207	30	116	8.3

Table 3: mChaff search run times of “randomized” Hole-n instances augmented with linear SBPs based on different sets of permutations. Tot denotes the total number of permutations used in constructing each SBP.

berg and Novikov, 2002] or GRASP [Silva and Sakallah, 1999] in the same experiment yielded opposite results as expected.

Table 2 offers additional data to evaluate symmetry-breaking by generators, which may not be complete. We added symmetry-breaking predicates built for pairwise products of generators, but the overall runtimes increased. While additional SBPs may break more symmetries, their overhead does not justify their use.

Table 3 describes experiments with generators that have long cycles in which we evaluated extensions to symmetry-breaking by generators. Namely, we tried adding powers of all generators and, alternatively, adding pairwise products of generators. Neither extension proved useful, which supports our main symmetry-breaking approach.

6 Conclusions

The main contribution of our work is a better construction of symmetry-breaking predicates for Boolean satisfiability. We empirically demonstrate improvements both in the size of predicates and the run time of SAT solvers after these predicates are added to the original CNF instances. We also show that (1) symmetry-breaking by generators is difficult to improve upon, and that (2) the efficiency of symmetry-breaking does not improve when larger cycles are found in generators.

Our work articulates that better symmetry finding algorithms would be useful, especially if tailored to CNF formulas and, perhaps, the kinds of symmetry groups commonly found in structured CNF instances.

Acknowledgments

This work was funded by the DARPA/MARCO Gigascale Silicon Research Center and by an Agere Systems/SRC Research fellowship.

References

- [Aloul *et al.*, 2002] F. Aloul, A. Ramani, I. Markov, and K. Sakallah, “Solving Difficult SAT Instances in the Presence of Symmetries,” in *Proc. Design Automation Conf. (DAC)*, pp. 731-736, 2002.
- [Crawford *et al.*, 1996] J. Crawford, M. Ginsberg, E. Luks, and A. Roy, “Symmetry-breaking predicates for search problems,” in *Proc. Intl. Conf. Principles of Knowledge Representation and Reasoning*, pp. 148-159, 1996.
- [Biere *et al.*, 1999] A. Biere, A. Cimatti, E. Clarke, M. Fujita, and Y. Zhu, “Symbolic Model Checking Using SAT Procedures Instead of BDDs,” in *Proc. Design Automation Conf. (DAC)*, pp. 317-320, 1999.
- [DIMACS] DIMACS Challenge benchmarks in <ftp://Dimacs.rutgers.EDU/pub/challenge/sat/benchmarks/cnf>.
- [Fraleigh, 2000] J. B. Fraleigh. A First Course in Abstract Algebra. 6th ed. Addison Wesley Longman, Reading, Massachusetts, 2000.
- [Goldberg and Novikov, 2002] E. Goldberg and Y. Novikov, “BerkMin: A fast and robust SAT-solver,” in *Proc. Design, Automation, and Test in Europe Conf. (DATE)*, pp. 142-149, 2002.
- [McKay, 1981] B. McKay, “Practical Graph Isomorphism,” in *Congressus Numerantium*, vol. 30, pp. 45-87, 1981. <http://cs.anu.edu.au/~bdm/naauty/>
- [Moskewicz *et al.*, 2001] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik, “Chaff: Engineering an Efficient SAT Solver,” in *Proc. Design Automation Conf. (DAC)*, pp. 530-535, 2001.
- [Nam *et al.*, 2001] G. Nam, F. Aloul, K. Sakallah, and R. Rutenbar, “A Comparative Study of Two Boolean Formulations of FPGA Detailed Routing Constraints,” in *Proc. Intl. Symp. on Physical Design (ISPD)*, pp. 222-227, 2001.
- [SAT 2002] SAT 2002 Competition, <http://www.satlive.org/SATCompetition/submittedbenchs.html>
- [Silva and Sakallah, 1999] J. Silva and K. Sakallah, “GRASP: A New Search Algorithm for Satisfiability,” in *IEEE Trans. on Computers*, 48(5), pp. 506-521, 1999.
- [Spitznagel, 1994] E. Spitznagel, “Review of Mathematical Software, GAP,” in *Notices Amer. Math. Soc.*, 41(7), pp. 780-782, 1994.
- [Stephan *et al.*, 1996] P. Stephan, R. Brayton, and A. Sangiovanni-Vincentelli, “Combinational Test Generation Using Satisfiability,” in *IEEE Trans. on Computer-Aided Design*, 15(9), pp. 1167-1175, 1996.
- [Urquhart, 1987] A. Urquhart, “Hard Examples for Resolution,” in *Journal of the ACM*, 34(1), pp. 209-219, 1987.
- [Velev and Bryant, 2001] M. N. Velev and R. E. Bryant, “Effective Use of Boolean Satisfiability Procedures in the Formal Verification of Superscalar and VLIW Microprocessors,” in *Proc. Design Automation Conf. (DAC)*, pp. 226-231, 2001.

Tractable Symmetry Breaking for CSPs with Interchangeable Values

P. Van Hentenryck
Department of Computer Science
Brown University, Box 1910
Providence, RI 02912

P. Flener, J. Pearson, and M. Ågren
Department of Information Technology
Uppsala University, Box 337
S-751 05 Uppsala, Sweden

Abstract

Symmetry breaking in CSPs has attracted considerable attention in recent years. Various general schemes have been proposed to eliminate symmetries during search. In general, these schemes may take exponential space or time to eliminate all symmetries. This paper studies classes of CSPs for which symmetry breaking is tractable. It identifies several CSP classes which feature various forms of value interchangeability and shows that symmetry breaking can be performed in constant time and space during search using dedicated search procedures. Experimental results also show the benefits of symmetry breaking on these CSPs, which encompass many practical applications.

1 Introduction

Symmetry breaking for constraint satisfaction problems has attracted considerable attention in recent years (See the last CP proceedings, the CP'01 and CP'02 workshops on symmetries, and [Crawford *et al.*, 1996; Freuder, 1991; Backofen and Will, 1999; Puget, 1993] for some less recent papers). Indeed, many applications naturally exhibit symmetries and symmetry breaking may drastically improve performance (e.g., [Barnier and Brisset, 2002; Meseguer and Torras, 2001; Puget, 2002]). An important contribution in this area has been the development of various general schemes for symmetry breaking in CSPs (e.g., SBDS [Gent and Smith, 2000] and SBDD [Fahle *et al.*, 2001; Focacci and Milano, 2001]). Unfortunately, these schemes, in general, may require exponential resources to break all symmetries. Indeed, some schemes require exponential space to store all the nogoods generated through symmetries, while others may take exponential time to discover whether a partial assignment is symmetric to one of the existing nogoods. As a consequence, practical applications often place limits on how many nogoods can be stored and/or which symmetries to break.

This paper approaches symmetry breaking from a different, orthogonal, standpoint. *Its goal is to identify classes of CSPs that are practically relevant and for which symmetry breaking is tractable, i.e., symmetry breaking is polynomial in time and space.* It identifies several such classes, which

encompass many practical applications. *These CSPs feature various forms of value interchangeability and the paper shows that symmetry breaking can be performed in constant time and space during search using dedicated search procedures.* The paper also reports preliminary experimental results indicating that symmetry breaking on these CSPs brings significant computational benefits. Finally, the paper introduces the new notions of existential and abstract nogoods, which were used to derive the results for the various CSP classes. We believe that these notions are helpful to derive many other classes of tractable symmetries. In particular, they also allowed us to derive classes of tractable *variable* symmetries, which we cannot include here for space reasons. As such, this paper should be viewed only as a first step in this fascinating area.

It is also useful to contrast our approach with the research avenue pioneered by [Freuder, 1991] on value interchangeability. Freuder also introduced various forms of value interchangeability. However, his goal was to *discover* symmetries inside CSPs and to reformulate them by preprocessing to remove symmetries. Unfortunately, discovering symmetries in CSPs is not tractable for many interesting classes of CSPs. *Our research, in contrast, assumes that modellers are aware of the symmetries in their applications.* It focuses on how to exploit this knowledge *during search* to eliminate symmetries efficiently. Consider, for instance, the scene-shooting problem featured in [Van Hentenryck, 2002]. This problem aims at producing a movie (or a series) at minimal cost by deciding when to shoot scenes. Each scene involves a number of actors and at most 5 scenes a day can be filmed. All actors of a scene must be present on the day the scene is shot. The actors have fees representing the amount to be paid per day they spend in the studio. The goal of the application is to minimize the production costs. An optimal solution to this problem can be modelled as an assignment of scenes to days which minimizes the production costs. It should be apparent that the exact days assigned to the scenes have no importance in this application and are fully interchangeable. What is important is how the scenes are clustered together. Our approach does not aim at discovering this fact; rather it focuses on how to exploit it to eliminate the symmetries it induces.

The rest of the paper is organized as follows. After some preliminaries, Sections 3, 4, and 5 study three classes of CSPs for which symmetry breaking is tractable. For space reasons,

only the first class is presented in full detail with proofs. Section 6 briefly reports some experimental results. Section 7 concludes the paper.

2 Preliminaries

This section defines the main concepts used in this paper. The definitions, although they capture the informal meaning of CSPs, are non-standard and simplify the definitions and proofs considerably. The basic idea is that the set of constraints is abstracted by a Boolean function which holds if all the constraints are satisfied (since we are not interested in the constraint structure). Solutions are also represented as functions (assignments) from variables to their sets of values.

Definition 2.1. A CSP is a triplet $\langle V, D, C \rangle$, where V denotes the set of variables, D denotes the set of possible values for these variables, and $C : (V \rightarrow D) \rightarrow \text{Bool}$ is a constraint that specifies which assignments of values to the variables are solutions. A solution to a CSP $\mathcal{P} = \langle V, D, C \rangle$ is a function $\sigma : V \rightarrow D$ such that $C(\sigma) = \text{true}$. The set of solutions to a CSP $\mathcal{P}$ is denoted by $\text{Sol}(\mathcal{P})$.

Algorithms to solve CSPs manipulate partial assignments. It is often important to reason about which variables are already assigned (the domain of the partial assignment) and the set of values they are assigned to (the image of the assignment). Note that domains represent sets of variables in this paper (not values as is traditional), since we use functions to model (partial) assignments.

Definition 2.2. Let $\mathcal{P} = \langle V, D, C \rangle$ be a CSP. A partial assignment for $\mathcal{P}$ is a function $\alpha : W \subseteq V \rightarrow D$. The domain of α , denoted by $\text{dom}(\alpha)$, is W . The image of α , denoted by $\text{image}(\alpha)$, is the set $\{\alpha(v) \mid v \in \text{dom}(\alpha)\}$. For each value $d \in \text{image}(\alpha)$, we use $\alpha^{-1}(d)$ to denote the set $\{v \mid v \in \text{dom}(\alpha) \wedge \alpha(v) = d\}$.

In this paper, we often denote a partial assignment α by a conjunction of equations

$$v_{i_1} = \alpha(v_{i_1}) \wedge \dots \wedge v_{i_k} = \alpha(v_{i_k})$$

where $\text{dom}(\alpha) = \{v_{i_1}, \dots, v_{i_k}\}$. For instance, the partial assignment $v_1 = 1 \wedge v_2 = 2 \wedge v_3 = 3$ represents the function whose domain is $\{v_1, v_2, v_3\}$ and which assigns the value i to v_i . We also denote the empty assignment by ϵ . The next four definitions define nogoods formally and when a (partial) assignment violates a nogood.

Definition 2.3. Let $\mathcal{P}$ be a CSP and α be a partial assignment for $\mathcal{P}$. A partial assignment θ is an extension of α for $\mathcal{P}$ if $\text{dom}(\alpha) \subseteq \text{dom}(\theta)$ and $\forall v \in \text{dom}(\alpha) : \theta(v) = \alpha(v)$.

Definition 2.4. Let $\mathcal{P} = \langle V, D, C \rangle$ be a CSP and α be a partial assignment for $\mathcal{P}$. A completion of α for $\mathcal{P}$ is an extension $\alpha' : V \rightarrow D$ of α for $\mathcal{P}$. The set of all completions of α for $\mathcal{P}$ is denoted by $\text{Comp}(\alpha, \mathcal{P})$.

Definition 2.5. Let $\mathcal{P}$ be a CSP. A nogood α for $\mathcal{P}$ is a partial assignment for $\mathcal{P}$ such that $\text{Comp}(\alpha, \mathcal{P}) \cap \text{Sol}(\mathcal{P}) = \emptyset$.

Definition 2.6. Let $\mathcal{P}$ be a CSP, α be a nogood for $\mathcal{P}$, and θ be a partial assignment for $\mathcal{P}$. θ violates nogood α for $\mathcal{P}$ if θ is an extension of α for $\mathcal{P}$.

Proposition 2.7. Let $\mathcal{P}$ be a CSP. If θ violates nogood α for $\mathcal{P}$, then $\text{Comp}(\theta, \mathcal{P}) \cap \text{Sol}(\mathcal{P}) = \emptyset$.

This last proposition shows that nogoods can be lifted from the children to their parent.

Proposition 2.8. Let $\mathcal{P} = \langle V, D, C \rangle$ be a CSP and $D = \{d_1, \dots, d_m\}$. Let α be a partial assignment for $\mathcal{P}$ with $\text{dom}(\alpha) = \{v_{i_1}, \dots, v_{i_k}\}$ and let every $\alpha_i = \alpha \wedge v_{i_k+1} = d_i$ ($1 \leq i \leq m$) be a nogood for $\mathcal{P}$. Then α is a nogood for $\mathcal{P}$.

3 Value-Interchangeable CSPs

There are many applications in resource allocation and scheduling where the exact values taken by the variables are not important. What is significant is which variables take the same values or, in other terms, how the variables are clustered. This notion of symmetry is what [Freuder, 1991] calls full interchangeability for all variables and all values. The prototypical example is graph-coloring. Another, more complex, example is the scene allocation application mentioned in the introduction. This section shows that symmetry breaking for these problems is tractable and takes constant time and space. We first define value-interchangeable CSPs.

Definition 3.1. Let $\mathcal{P} = \langle V, D, C \rangle$ be a CSP. $\mathcal{P}$ is value-interchangeable if, for each solution $\sigma \in \text{Sol}(\mathcal{P})$ and each bijection $b : D \rightarrow D$, the function $b \circ \sigma \in \text{Sol}(\mathcal{P})$.

In the following, we use ICSP as an abbreviation for value-interchangeable CSP. The following theorem is a fundamental characterization of nogoods for ICSPs.

Theorem 3.2. Let $\mathcal{P} = \langle V, D, C \rangle$ be an ICSP, let α be a nogood for $\mathcal{P}$, and let $b : D \rightarrow D$ be a bijection. Then $b \circ \alpha$ is a nogood.

The closure of a nogood α for an ICSP captures all the “symmetric” nogoods that can be obtained from α .

Definition 3.3. Let $\mathcal{P} = \langle V, D, C \rangle$ be an ICSP and α be a nogood for $\mathcal{P}$. The closure of α for $\mathcal{P}$, denoted by $\text{Closure}(\alpha, \mathcal{P})$, is the set $\{b \circ \alpha \mid b : D \rightarrow D \text{ is a bijection}\}$.

We now show that the closure of a nogood can be characterized compactly and that membership to the closure of a nogood can be tested in linear time for ICSPs. We first introduce the concept of existential nogood, which simplifies the proofs and intuitions.

Definition 3.4. Let $\mathcal{P} = \langle V, D, C \rangle$ be an ICSP and α be a nogood for $\mathcal{P}$. Let $\text{image}(\alpha) = \{d_1, \dots, d_k\}$. The existential nogood of α for $\mathcal{P}$, denoted by $\text{Enogood}(\alpha, \mathcal{P})$, is the set of all functions $\gamma : \text{dom}(\alpha) \rightarrow D$ satisfying

$$\exists e_1, \dots, e_k \in D :$$

$$\forall i \in 1..k : \forall v_j \in \alpha^{-1}(d_i) : \gamma(v_j) = e_i \wedge \text{alldiff}(e_1, \dots, e_k).$$

The following lemma indicates that an existential nogood precisely captures the closure of a nogood.

Lemma 3.5. Let $\mathcal{P}$ be an ICSP and α be a nogood for $\mathcal{P}$. Then $\text{Enogood}(\alpha, \mathcal{P}) = \text{Closure}(\alpha, \mathcal{P})$.

Proof. Let $\text{image}(\alpha) = \{d_1, \dots, d_k\}$. By definition of the image, we have that

$$\forall i \in 1..k : \forall v_j \in \alpha^{-1}(d_i) : \alpha(v_j) = d_i \wedge \text{alldiff}(d_1, \dots, d_k).$$

We first show that $Closure(\alpha, \mathcal{P}) \subseteq Enogood(\alpha, \mathcal{P})$. Let $\gamma \in Closure(\alpha, \mathcal{P})$. This means that there exists a bijection b such that $\gamma = b \circ \alpha$. Thus, γ satisfies

$$\forall i \in 1..k : \forall v_j \in \alpha^{-1}(d_i) : \gamma(v_j) = b(d_i) \wedge \\ alldiff(b(d_1), \dots, b(d_k))$$

by definition of a bijection, and hence $\gamma \in Enogood(\alpha, \mathcal{P})$. We now show that $Enogood(\alpha, \mathcal{P}) \subseteq Closure(\alpha, \mathcal{P})$. Let $\delta \in Enogood(\alpha, \mathcal{P})$. There exist some values $a_1, \dots, a_k \in D$ such that

$$\forall i \in 1..k : \forall v_j \in \alpha^{-1}(d_i) : \delta(v_j) = a_i \wedge alldiff(a_1, \dots, a_k).$$

Since $a_1, \dots, a_k$ are all different, there exists a bijection b satisfying $\forall i \in 1..k : b(d_i) = a_i$. Hence δ can be rewritten as $b \circ \alpha$ and $\delta \in Closure(\alpha, \mathcal{P})$. $\square$

It is not obvious that membership to an existential nogood can be tested efficiently, since it involves an existential quantification. However, due to the nature of the underlying constraints, it is easy to eliminate the existential variables by selecting a representative for each set $\alpha^{-1}(d_i)$. This is precisely the motivation underlying the concept of abstract nogoods defined below. Consider the existential formula

$$\exists e_1, e_2, e_3 \in D : \theta(v_1) = e_1 \wedge \theta(v_2) = e_2 \wedge \theta(v_3) = e_3 \wedge \\ \theta(v_4) = e_1 \wedge \theta(v_5) = e_2 \wedge alldiff(e_1, e_2, e_3).$$

The variables e_1, e_2 , and e_3 can be eliminated to produce

$$\theta(v_1) = \theta(v_4) \wedge \theta(v_2) = \theta(v_5) \wedge alldiff(\theta(v_1), \theta(v_2), \theta(v_3)).$$

Definition 3.6. Let $\mathcal{P} = \langle V, D, C \rangle$ be an ICSP and α be a nogood for $\mathcal{P}$. Let $image(\alpha) = \{d_1, \dots, d_k\}$ and let

$$v_{r_i} \in \alpha^{-1}(d_i) \quad (1 \leq i \leq k).$$

The *abstract nogood* of α wrt $\mathcal{P}$, denoted by $Anogood(\alpha, \mathcal{P})$, is the set of functions $\gamma : dom(\alpha) \rightarrow D$ satisfying

$$\forall i \in 1..k : \forall v_j \in \alpha^{-1}(d_i) : \gamma(v_j) = \gamma(v_{r_i}) \wedge \\ alldiff(\gamma(v_{r_1}), \dots, \gamma(v_{r_k})).$$

Lemma 3.7. Let $\mathcal{P}$ be an ICSP and α be a nogood for $\mathcal{P}$. Then $Enogood(\alpha, \mathcal{P}) = Anogood(\alpha, \mathcal{P})$.

For simplicity, we will often denote the abstract nogood condition in terms of global constraints

$$\forall i \in 1..k : allequal(v_j \in \alpha^{-1}(d_i)) \gamma(v_j) \wedge \\ alldiff(\gamma(v_{r_1}), \dots, \gamma(v_{r_k}))$$

where $allequal(i \in S) a_i$ holds if all the a_i are the same value. It can be seen that there exists a linear time algorithm to test whether a partial assignment θ violates a nogood in $Closure(\alpha, \mathcal{P})$. It suffices to test whether θ satisfies the formula above whenever $dom(\alpha) \subseteq dom(\theta)$.

Lemma 3.8. Let $\mathcal{P}$ be an ICSP, α be a nogood for $\mathcal{P}$, and θ be a partial assignment. There exists a linear-time algorithm that tests whether θ violates any nogood in $Closure(\alpha, \mathcal{P})$.

Proof. Direct consequence of lemmas 3.5 and 3.7 and the fact that the abstract nogood is linear in $|V|$. $\square$

Lemma 3.9. Let $\mathcal{P} = \langle V, D, C \rangle$ be an ICSP, $D = \{d_1, \dots, d_m\}$, and α be a partial assignment for $\mathcal{P}$ with $dom(\alpha) = \{v_{i_1}, \dots, v_{i_k}\}$. Let every $\alpha_i = \alpha \wedge v_{i_{k+1}} = d_i$ ($1 \leq i \leq m$) be a nogood for $\mathcal{P}$. Then,

1. α is a nogood for $\mathcal{P}$;
2. if θ violates a nogood in $\bigcup_{i=1}^m Closure(\alpha_i, \mathcal{P})$, then θ violates a nogood in $Closure(\alpha, \mathcal{P})$.

Lemmas 3.8 and 3.9 indicate that symmetry breaking is tractable for ICSPs. Lemma 3.9 indicates that abstract nogoods are needed only for frontier nodes of the search tree (i.e., closed nodes whose parents are open). Once its children are explored, the abstract nogood of the parent subsumes the abstract nogoods of its children. Hence, maintaining the nogood takes space $O(|F||V|)$ if F is the set of frontier nodes. We formalize this result using variable decomposition trees.

Definition 3.10. A *variable decomposition tree* for a CSP $\mathcal{P} = \langle V, D, C \rangle$, where $D = \{d_1, \dots, d_m\}$, is a search tree where nodes represent partial assignments for $\mathcal{P}$ and nodes are decomposed as follows: given a node representing a partial assignment α , where $dom(\alpha) = \{v_{i_1}, \dots, v_{i_k}\}$, its children represent the partial assignments $\alpha_i = \alpha \wedge v_{i_{k+1}} = d_i$ ($1 \leq i \leq m$) for some variable $v_{i_{k+1}} \in V \setminus dom(\alpha)$.

Note that variable decomposition trees capture both static and dynamic variable orderings.

Theorem 3.11. Let $\mathcal{P}$ be an ICSP and let F be the set of frontier nodes in a variable decomposition search tree for $\mathcal{P}$. Symmetry breaking for $\mathcal{P}$ requires $O(|F||V|)$ space for storing the nogoods. In addition, testing if a partial assignment θ violates a nogood takes $O(|F||V|^2)$ time in the worst case.

The result above can be strengthened considerably in fact. We will show that search procedures exploring a variable decomposition tree can remove all symmetries in constant time and space. Before presenting the theoretical results, we illustrate the intuition using depth-first search. The basic intuition comes from the structure of the abstract nogoods. Consider the partial assignment

$$v_1 = 1 \wedge v_2 = 2 \wedge v_3 = 3 \wedge v_4 = 1 \wedge v_5 = 2$$

and assume that depth-first search tries to assign variable v_6 and that the set of possible values is $1..10$. The failure of $v_6 = 1$ produces an abstract nogood θ which holds if

$$allequal(\theta(v_1), \theta(v_4), \theta(v_6)) \wedge \\ allequal(\theta(v_2), \theta(v_5)) \wedge \\ alldiff(\theta(v_1), \theta(v_2), \theta(v_3))$$

Since $v_1, \dots, v_5$ remain instantiated when the next value is tried for v_6 , the abstract nogood for this part of this next branch holds if $\theta(v_6) = 1$ imposing that v_6 be assigned a value different from 1. The assignments $v_6 = 2$ and $v_6 = 3$ produce similar abstract nogoods. Now observe what happens for an assignment of a value $i \in 4..10$, say 6. The abstract nogood θ is now defined as

$$allequal(\theta(v_1), \theta(v_4)) \wedge \\ allequal(\theta(v_2), \theta(v_5)) \wedge \\ alldiff(\theta(v_1), \theta(v_2), \theta(v_3), \theta(v_6))$$

which can be partially evaluated to $alldiff(1, 2, 3, \theta(v_6))$. The disjunction of all these nogoods can be partially evaluated to

```

bool ILABEL( $\mathcal{P}$ ) { return ILABELA( $\mathcal{P}, \epsilon$ ); }
bool ILABELA( $\langle V, D, C \rangle, \theta$ ) {
  if  $\text{dom}(\theta) = V$  then return  $C(\theta)$ ;
  select  $v$  in  $V \setminus \text{dom}(\theta)$ ;
   $A := \text{image}(\theta)$ ;
  if  $\text{image}(\theta) \neq D$  then
    select  $f$  in  $D \setminus \text{image}(\theta)$ ;  $A := A \cup \{f\}$ ;
    forall ( $d \in A$ )
       $\theta' := \theta \wedge v = d$ ;
      if  $\neg \text{Failure}(\langle V, D, C \rangle, \theta')$  then
        if ILABELA( $\langle V, D, C \rangle, \theta'$ ) then
          return true;
    return false;
}

```

Figure 1: A Labeling Procedure for ICSPs.

$\theta(v_6) = 1 \vee \theta(v_6) = 2 \vee \theta(v_6) = 3 \vee \text{alldiff}(1, 2, 3, \theta(v_6))$

which cannot be satisfied by any assignment to v_6 . It follows that v_6 must only be assigned to the previously assigned values 1, 2, 3 or to exactly one new value in 4..10. In other words, in a variable decomposition tree, only some of the children need to be explored: those which assign variable $v_{i_{k+1}}$ to a value in $\text{dom}(\alpha)$ and exactly one other child. Note that this result is independent from the set of constraints. It is the essence of the labeling procedure for graph-coloring in [Kubale and Jackowski, 1985] and in the scene allocation problem in [Van Hentenryck, 2002]. This procedure, which breaks all symmetries for ICSPs, is formalized in Figure 1. It uses the function $\text{Failure}(\mathcal{P}, \theta)$ ($\mathcal{P} = \langle V, D, C \rangle$) which satisfies the property

$$\text{Failure}(\mathcal{P}, \theta) \Rightarrow [\forall \beta \in \text{Comp}(\theta, \mathcal{P}) : \neg C(\beta)].$$

To our knowledge, this paper provides the first formal proof that these algorithms break all the symmetries in their respective problems. To prove the correctness of ILABEL, and other related search procedures, it is useful to introduce the concept of compact variable decomposition tree.

Definition 3.12. A compact variable decomposition tree for an ICSP $\mathcal{P} = \langle V, D, C \rangle$, where $D = \{d_1, \dots, d_m\}$, is a search tree where nodes represent partial assignments for $\mathcal{P}$ and nodes are decomposed as follows: given a node representing a partial assignment α , where $\text{dom}(\alpha) = \{v_{i_1}, \dots, v_{i_k}\}$, for some variable $v_{i_{k+1}} \in V \setminus \text{dom}(\alpha)$, there are children representing the partial assignments $\alpha_i = \alpha \wedge v_{i_{k+1}} = d_o$ for all $d_o \in \text{image}(\alpha)$ and exactly one child representing a partial assignment $\alpha_i = \alpha \wedge v_{i_{k+1}} = d_n$ for some $d_n \in D \setminus \text{image}(\alpha)$, if $D \setminus \text{image}(\alpha)$ is not empty.

The following lemma, which states that compact variable decomposition trees are complete, follows directly from the examination of the nogoods above.

Lemma 3.13. Let $\mathcal{P} = \langle V, D, C \rangle$ be an ICSP and S be all the complete assignments in a variable decomposition tree for $\mathcal{P}$. Then, the closure $\{b \circ \alpha \mid \alpha \in S \wedge b : D \rightarrow D \text{ is a bijection}\}$ is equal to $\text{Sol}(\mathcal{P})$.

Our next lemma states that a compact variable decomposition tree never matches any nogood generated during search.

Lemma 3.14. Let $\mathcal{P}$ be an ICSP and T be a compact variable decomposition tree for $\mathcal{P}$. The partial assignment α of a node in T never matches any nogood generated during the exploration of T (except the one it possibly generates).

Proof. By Lemma 3.9, it suffices to show that a partial assignment θ never matches a nogood generated by its siblings or the siblings of one of its ancestors in the tree. The proof is by induction on the depth of the tree. At the depth of θ , the result follows from the inspections of the nogood as discussed earlier. Consider a depth $dp' < dp$ and a nogood α generated by one of the left or right branches at that depth. We can restrict attention to the projection of θ to the variables instantiated at that depth, i.e., we can restrict attention to $\theta' : \text{dom}(\alpha) \rightarrow D$ satisfying $\forall v \in \text{dom}(\alpha) : \theta'(v) = \theta(v)$.

We show that $\theta' \notin \text{Closure}(\alpha, \mathcal{P})$. Let v be the variable assigned at depth d' . Observe that there exists a partial assignment α' such that $\alpha = \alpha' \wedge v = e_1$ and $\theta' = \alpha' \wedge v = e_2$, for some $e_1, e_2 \in D$ and $e_1 \neq e_2$. By definition of a compact variable decomposition tree, the values e_1 and e_2 must belong to $\text{image}(\alpha') \cup \{d\}$, where $d \notin \text{image}(\alpha')$.

Consider the case where $e_1, e_2 \in \text{image}(\alpha')$. That means that there exist v_i and v_j in $\text{dom}(\alpha')$ such that $\alpha(v) = \alpha(v_i) \neq \alpha(v_j)$ and $\theta'(v) = \theta'(v_j) \neq \theta'(v_i)$. If $\theta' \in \text{Closure}(\alpha, \mathcal{P})$, it can be rewritten as $\theta' = b \circ \alpha$ for some bijection b . Hence $\theta'(v) = \theta'(v_i)$, which is impossible.

Assume now that $e_1 \in \text{image}(\alpha')$ and $e_2 = d$. It follows that $\alpha(v) = \alpha(v_i)$ for some $v_i \in \text{dom}(\alpha')$ and that $\theta'^{-1}(d) = \{v\}$. If $\theta' \in \text{Closure}(\alpha, \mathcal{P})$, then $\theta' = b \circ \alpha$ for some bijection b . Hence $\theta'(v) = \theta'(v_i)$ which is impossible.

Assume that $e_2 \in \text{image}(\alpha')$ and $e_1 = d$. Then, $\alpha^{-1}(d) = \{v\}$ and $\theta'(v) = \theta'(v_i)$ for some $v_i \in \text{dom}(\alpha')$. If $\theta' \in \text{Closure}(\alpha, \mathcal{P})$, then $\theta' = b \circ \alpha$ for some bijection b . Hence, $\theta'(v) = b(d) \wedge \theta'^{-1}(b(d)) = \{v\}$, which is impossible since $\theta'(v) = \theta'(v_i)$ for some $v_i \in \text{dom}(\alpha')$. $\square$

The correctness of Procedure ILABEL follows from the two lemmas above. Other search strategies, e.g., LDS, also remove all symmetries in constant time and space.

Theorem 3.15. Procedure ILABEL breaks all the symmetries in constant time and constant space for an ICSP $\mathcal{P}$, i.e., it never matches any member of the closure of any nogood generated during search.

4 Piecewise-Interchangeable CSPs

In many applications, e.g., in resource allocation and scheduling, the values are taken from disjoint sets but they are interchangeable in each set. For instance, in the scene allocation problem, we can easily imagine a version of the problem where days are divided in morning and afternoon sessions. Actors probably have strong preferences (and thus different fees for the morning and afternoon sessions) but the day of the session may not matter. This paper captures this (more limited) form of interchangeability by piecewise interchangeable CSPs. We first define this class of CSPs formally and then derive similar results as for ICSPs. We state the main definitions and theorems only, since the derivation is similar to the one for ICSPs. As traditional, we use $D_1 + D_2$ to denote the disjoint union of D_1 and D_2 . Our definitions and results only

consider two distinct sets of values, for simplicity. It is easy to generalize them to an arbitrary number of sets.

Definition 4.1. Let D_1 and D_2 be two disjoint sets. A *piecewise bijection* $b : D_1 + D_2 \rightarrow D_1 + D_2$ is a bijection defined as $b(d) = b_i(d)$ if $d \in D_i$ where $b_i : D_i \rightarrow D_i$ is a bijection ($1 \leq i \leq 2$).

Definition 4.2. Let $\mathcal{P} = \langle V, D_1 + D_2, C \rangle$ be a CSP. $\mathcal{P}$ is *piecewise value-interchangeable* if, for each solution $\sigma \in \text{Sol}(\mathcal{P})$ and each piecewise bijection $b : D_1 + D_2 \rightarrow D_1 + D_2$, the function $b \circ \sigma \in \text{Sol}(\mathcal{P})$.

In the following, we use PICSP as an abbreviation for piecewise value-interchangeable CSP.

Definition 4.3. Let $\mathcal{P} = \langle V, D_1 + D_2, C \rangle$ be a PICSP and α be a nogood for $\mathcal{P}$. The *closure* of α for $\mathcal{P}$, denoted by $\text{Closure}(\alpha, \mathcal{P})$, is the set $\{b \circ \alpha \mid b : D_1 + D_2 \rightarrow D_1 + D_2 \text{ is a piecewise bijection}\}$.

We now specify the abstract nogoods for PICSPs. The key intuition is to separate the values from D_1 and D_2 .

Definition 4.4. Let $\mathcal{P} = \langle V, D_1 + D_2, C \rangle$ be a PICSP and α be a nogood for $\mathcal{P}$. Let $\text{image}(\alpha) = \{d_1^1, \dots, d_k^1, d_1^2, \dots, d_l^2\}$, where $d_i^1 \in D_1, d_i^2 \in D_2$, and let

$$\begin{aligned} v_{r_i^1} &\in \alpha^{-1}(d_i^1) \quad (1 \leq i \leq k); \\ v_{r_i^2} &\in \alpha^{-1}(d_i^2) \quad (1 \leq i \leq l). \end{aligned}$$

The *abstract nogood* of α wrt $\mathcal{P}$, denoted by $\text{Anogood}(\alpha, \mathcal{P})$, is the set of functions $\gamma : \text{dom}(\alpha) \rightarrow D_1 + D_2$ satisfying

$$\begin{aligned} \forall i \in 1..k : & \text{allegal}(\gamma(v_{r_i^1}) \in \alpha^{-1}(d_i^1)) \gamma(v_j) \wedge \\ \forall i \in 1..k : & \forall v_j \in \alpha^{-1}(d_i^1) : v_j \in D_1 \wedge \\ & \text{alldiff}(\gamma(v_{r_1^1}), \dots, \gamma(v_{r_k^1})) \wedge \\ \forall i \in 1..l : & \text{allegal}(\gamma(v_{r_i^2}) \in \alpha^{-1}(d_i^2)) \gamma(v_j) \wedge \\ \forall i \in 1..l : & \forall v_j \in \alpha^{-1}(d_i^2) : v_j \in D_2 \wedge \\ & \text{alldiff}(\gamma(v_{r_1^2}), \dots, \gamma(v_{r_l^2})). \end{aligned}$$

Figure 2 depicts a labeling procedure **PILABEL** for PICSPs, which breaks all value symmetries in constant time and space. Procedure **PILABEL** generalizes **ILABEL** by considering the already assigned values in both sets D_1 and D_2 , as well as one new value (if any) from both sets.

Theorem 4.5. Procedure **PILABEL** breaks all the symmetries in constant time and constant space for PICSPs.

5 Wreath-Value Interchangeable CSPs

We now consider another, more complex, class of CSPs, which assigns a pair of values (d_1, d_2) from a set $D_1 \times D_2$ to each variable v . Values in D_1 are fully interchangeable and, for a fixed value in D_1 , the values in D_2 are fully interchangeable as well. These problems are called wreath value-interchangeable in this paper, because the symmetry group corresponds to a wreath product of symmetry groups [Cameron, 1999]. Such problems arise naturally in a variety of applications in resource allocation and scheduling. Consider, for instance, the problem of scheduling a meeting where different groups must meet some day of the week in some room subject to constraints. The days are fully interchangeable and, on a given day, the rooms are fully interchangeable. Symmetry breaking for a wreath value-interchangeable CSP is tractable.

```

bool PILABEL( $\mathcal{P}$ ) { return PILABEL( $\mathcal{P}, \epsilon$ ); }
bool PILABEL( $\langle V, D_1 + D_2, C \rangle, \theta$ ) {
  if  $\text{dom}(\theta) = V$  then return  $C(\theta)$ ;
  select  $v$  in  $V \setminus \text{dom}(\theta)$ ;
   $A_1 := \text{image}(\theta) \cap D_1$ ;  $A_2 := \text{image}(\theta) \cap D_2$ ;
  if  $\text{image}(\theta) \cap D_1 \neq D_1$  then
    select  $f$  in  $D_1 \setminus \text{image}(\theta)$ ;  $A_1 := A_1 \cup \{f\}$ ;
  if  $\text{image}(\theta) \cap D_2 \neq D_2$  then
    select  $f$  in  $D_2 \setminus \text{image}(\theta)$ ;  $A_2 := A_2 \cup \{f\}$ ;
  forall( $d \in A_1 \cup A_2$ )
     $\theta' := \theta \wedge v = d$ ;
    if  $\neg \text{Failure}(\langle V, D_1 + D_2, C \rangle, \theta')$  then
      if PILABEL( $\langle V, D_1 + D_2, C \rangle, \theta'$ ) then
        return true;
  return false;
}

```

Figure 2: A Labeling Procedure for PICSPs.

Definition 5.1. Let D_1 and D_2 be two sets. A *wreath bijection* $b : D_1 \times D_2 \rightarrow D_1 \times D_2$ is a bijection defined as

$$b(\langle d_1, d_2 \rangle) = \langle b_1(d_1), b_2^{d_1}(d_2) \rangle$$

where $b_1 : D_1 \rightarrow D_1$ is a bijection and $b_2^{d_1} : D_2 \rightarrow D_2$ ($d_1 \in D_1$) is a bijection.

Definition 5.2. Let $\mathcal{P} = \langle V, D_1 \times D_2, C \rangle$ be a CSP. $\mathcal{P}$ is *wreath value-interchangeable* if, for each solution $\sigma \in \text{Sol}(\mathcal{P})$ and each wreath bijection $b : D_1 \times D_2 \rightarrow D_1 \times D_2$, the function $b \circ \sigma \in \text{Sol}(\mathcal{P})$.

In the following, we use WICSP as an abbreviation for wreath value-interchangeable CSP. We also use the following notations. If $d = (d_1, d_2)$ is a pair, $d[1] = d_1$ and $d[2] = d_2$. If T is a set of tuples, $T[i]$ denotes the set $\{d[i] \mid d \in T\}$ and $\text{filter}(T, i, d_i)$ denotes the set $\{d \mid d \in T \wedge d[i] = d_i\}$. If $\alpha : D_1 \times D_2 \rightarrow D_1 \times D_2$ is an assignment, $\alpha^{-1}((d_1, D_2))$ denotes the set $\{\alpha^{-1}((d_1, d_2)) \mid d_2 \in D_2\}$.

Definition 5.3. Let $\mathcal{P} = \langle V, D_1 \times D_2, C \rangle$ be a WICSP and α be a nogood for $\mathcal{P}$. The *closure* of α for $\mathcal{P}$, denoted by $\text{Closure}(\alpha, \mathcal{P})$, is the set $\{b \circ \alpha \mid b : D_1 \times D_2 \rightarrow D_1 \times D_2 \text{ is a wreath bijection}\}$.

We now specify abstract nogoods for WICSPs.

Definition 5.4. Let $\mathcal{P} = \langle V, D_1 \times D_2, C \rangle$ be a WICSP and α be a nogood for $\mathcal{P}$. Let $\text{image}(\alpha)[1] = \{d_1, \dots, d_k\}$, let $\text{filter}(\text{image}(\alpha), 1, d_i) = \{d_1^i, \dots, d_{l_i}^i\}$, and let

$$\begin{aligned} v_{r_i} &\in \alpha^{-1}(d_i, D_2) \quad (1 \leq i \leq k), \\ v_{r_j^i} &\in \alpha^{-1}((d_i, d_j)) \quad (1 \leq i \leq k, 1 \leq j \leq l_i). \end{aligned}$$

The *abstract nogood* of α wrt $\mathcal{P}$, denoted by $\text{Anogood}(\alpha, \mathcal{P})$, is the set of functions $\gamma : \text{dom}(\alpha) \rightarrow D_1 \times D_2$ satisfying

$$\begin{aligned} & \left[\forall i \in 1..k : \text{allegal}(\gamma(v_{r_i}) \in \alpha^{-1}((d_i, D_2))) \gamma(v_j)[1] \wedge \right. \\ & \quad \left. \text{alldiff}(\gamma(v_{r_1}), \dots, \gamma(v_{r_k}))[1] \right] \wedge \\ & \left[\forall i \in 1..l_1 : \text{allegal}(\gamma(v_{r_i^1}) \in \alpha^{-1}(d_1, d_1^1)) \gamma(v_j)[2] \wedge \right. \\ & \quad \left. \text{alldiff}(\gamma(v_{r_1^1}), \dots, \gamma(v_{r_{l_1}^1}))[1] \right] \wedge \\ & \dots \\ & \left[\forall i \in 1..l_k : \text{allegal}(\gamma(v_{r_i^k}) \in \alpha^{-1}(d_k, d_k^k)) \gamma(v_j)[2] \wedge \right. \\ & \quad \left. \text{alldiff}(\gamma(v_{r_1^k}), \dots, \gamma(v_{r_{l_k}^k}))[1] \right]. \end{aligned}$$

```

bool WILABEL( $\mathcal{P}$ ) { return WILABELA( $\mathcal{P}, \epsilon$ ); }
bool WILABELA( $\langle V, D_1 \times D_2, C \rangle, \theta$ ) {
  if  $\text{dom}(\theta) = V$  then return  $C(\theta)$ ;
  select  $v$  in  $V \setminus \text{dom}(\theta)$ ;
   $A_1 := \text{image}(\theta)[1]$ ;
  if  $A_1 \neq D_1$  then
    select  $f$  in  $D_1 \setminus A_1$ ;  $A_1 := A_1 \cup \{f\}$ ;
  forall ( $d_1 \in A_1$ )
     $A_2 := \text{filter}(\text{image}(\alpha), 1, d_1)[2]$ ;
    if  $A_2 \neq D_2$  then
      select  $f$  in  $D_2 \setminus A_2$ ;  $A_2 := A_2 \cup \{f\}$ ;
    forall ( $d_2 \in A_2$ )
       $\theta' := \theta \wedge v = (d_1, d_2)$ ;
      if  $\neg \text{Failure}(\langle V, D_1 \times D_2, C \rangle, \theta')$  then
        if WILABELA( $\langle V, D_1 \times D_2, C \rangle, \theta'$ ) then
          return true;
  return false;
}

```

Figure 3: A Labeling Procedure for WICSPs.

Theorem 5.5. *Procedure WILABEL breaks all the symmetries in constant time and constant space for a WICSP.*

6 Experimental Results

This section gives some preliminary results showing the benefits of symmetry breaking on these classes of problems. Table 1 gives results on graph coloring, an ICSP, and on partitioned graph coloring, a PICSP. In partitioned graph coloring, the colors are divided in 4 groups and are fully interchangeable in each group. The constraints express the usual graph-coloring property: no two adjacent vertices are colored with the same color. Table 1 gives the result of coloring graphs with 50% edge density with at most $n/(2 * \log_2(n))$ colors, where n is the number of nodes. The first column gives the number of nodes, the second column gives the time to color the graphs using the default labeling procedure, and the third and fourth columns report the time to color the graphs and the partitioned graphs with symmetry breaking. Note that the default labeling does not exploit symmetries and hence its time is similar for both graph coloring and partitioned graph coloring (second column). The last column gives the ratio of PILABEL and the default labeling procedure (the gains are too large for coloring to be reported as a ratio). All results are an average of 5 runs in milliseconds under SICStus-Prolog. They clearly show that symmetry breaking on these CSPs brings significant benefits.

7 Conclusion

This paper studied three classes of CSPs for which symmetry breaking is tractable. These CSP classes feature various forms of value interchangeability and allow symmetry breaking to be performed in constant time and space during search. Experimental results also show the benefits of symmetry breaking on these CSPs, which encompass many practical applications. There are many directions for future research. Of particular interest is the study of tractable classes of CSPs exhibiting variable symmetries where the set V has a

n	DL	$ILABEL$	$PILABEL$	%PILABEL
70	492	4	154	0.31
72	16418	96	4244	0.26
74	30790	34	18576	0.60
76	10622	24	2616	0.25
78	17786	26	15616	0.87
80	45268	86	9736	0.22
82	22946	46	15502	0.67
83	33206	42	9970	0.30

Table 1: Experimental Results on ICSPs and PICSPs.

complex structure (e.g., a Cartesian product). There are several classes of such CSPs for which symmetry breaking is tractable, although more complex. Finding effective search procedures for these CSPs is also a challenging problem.

8 Acknowledgments

Special thanks to the reviewers for their very interesting comments. All authors are partly supported by a STINT institutional grant IG2001-67 of STINT. P. Van Hentenryck is partly supported by an NSF ITR Award DMI-0121495. The Sweden-based authors are also supported by grant 221-99-369 of the Swedish Research Council.

References

- [Backofen and Will, 1999] Rolf Backofen and Sebastian Will. Excluding symmetries in constraint-based search. In CP'99.
- [Barnier and Brisset, 2002] N. Barnier and P. Brisset. Solving the Kirkman's schoolgirl problem in a few seconds. In CP'2002.
- [Cameron, 1999] P. Cameron. Permutation Groups, Cambridge Univ. Press, 1999.
- [Crawford et al., 1996] J. Crawford et al. Symmetry breaking predicates for search problems. In KR'96.
- [Fahle et al., 2001] Schamberger Fahle, Torsten, Stefan, and Meinolf Sellmann. Symmetry breaking. In CP'2001.
- [Focacci and Milano, 2001] Filippo Focacci and Michaela Milano. Global cut framework for removing symmetries. In CP'2001.
- [Freuder, 1991] E. Freuder. Eliminating interchangeable values in constraint satisfaction problems. In *Proc. AAAI'91*.
- [Gent and Smith, 2000] Ian P. Gent and B. Smith. Symmetry breaking during search in constraint programming. In ECAI'2000.
- [Kubale and Jackowski, 1985] M. Kubale and D. Jackowski. A Generalized Implicit Enumeration Algorithm for Graph Coloring. *CACM*, 28(4):412–418, 1985.
- [Meseguer and Torras, 2001] P. Meseguer and C. Torras. Exploiting symmetries within constraint satisfaction search. *Artificial Intelligence*, 129(1-2):133–163, 2001.
- [Puget, 1993] J.-F. Puget. On the satisfiability of symmetrical constrained satisfaction problems. In ISMIS'93.
- [Puget, 2002] J.-F. Puget. Symmetry Breaking Revisited. In CP'2002.
- [Van Hentenryck, 2002] P. Van Hentenryck. Constraint and Integer Programming in OPL. *Inform Journal on Computing*, 14(4):345–372, 2002.

DECISION THEORY

On the Foundations of *Expected Expected* Utility

Craig Boutilier

Department of Computer Science
University of Toronto
Toronto, ON, M5S 3H5, CANADA
cebly@cs.toronto.edu

Abstract

Intelligent agents often need to assess user utility functions in order to make decisions on their behalf, or predict their behavior. When uncertainty exists over the precise nature of this utility function, one can model this uncertainty using a distribution over utility functions. This view lies at the core of games with incomplete information and, more recently, several proposals for incremental preference elicitation. In such cases, decisions (or predicted behavior) are based on computing the *expected expected* utility (EEU) of decisions with respect to the distribution over utility functions. Unfortunately, decisions made under EEU are sensitive to the precise representation of the utility function. We examine the conditions under which EEU provides for sensible decisions by appeal to the foundational axioms of decision theory. We also discuss the impact these conditions have on the enterprise of preference elicitation more broadly.

1 Introduction

Most work on the foundations of decision theory—specifically on the justification of expected utility—has focused on *personal decision making*, that is, settings where a decision is being made by the “holder” of the utility function. Of course the decision maker may not be fully aware of (or have fully articulated) her utility function. The process of articulation is complex, and much work in decision analysis deals with preference elicitation and decision framing to help the decision maker formulate her decision problem [11]. However, this work is primarily concerned with eliciting enough information about preference tradeoffs to allow an (approximately) optimal decision to be made. While an analyst can never be sure about the true nature of the decision maker’s utility function, this uncertainty is not generally characterized explicitly, though its impact is often minimized through sensitivity analysis and related techniques.

Recent emphasis has been placed on the development of automated decision tools, where a decision is being made on behalf of a user whose utility function is imprecisely known. As in goal programming or other forms of interactive optimization, a space of possible utility functions is

usually maintained (often by imposing constraints on trade-off weights). A decision can be made based on this set of feasible utility functions. For example, Pareto optimal decisions can be identified [21; 18], or models based on min-max regret can be used to choose a specific decision [11; 2; 20]. In each of these models, the uncertainty regarding the utility function is characterized by the feasible utility set.

Somewhat less common is work in which the system’s uncertainty about a user’s utility function is quantified probabilistically. Some recent examples include [5; 6; 1]. In this work, a distribution over utility functions is assumed. The expected utility of a decision is determined not just by taking expectation over the outcomes of that decision, but also expectation over the space of possible utility functions. We use the term *expected expected utility* (EEU) to denote the value of a decision computed in this way. Elicitation strategies can be informed using the current distribution over utility functions. For example, value of information can be used to determine whether the improvement in decision quality given by a piece of information outweighs the cost of obtaining that information. Thus, characterizing one’s uncertainty over possible utility functions in a probabilistic fashion, and using EEU to determine decision quality, has much to recommend it from the point of view of elicitation.

Decision making using distributions over utility functions has been considered in other contexts. For example, Cyert and de Groot consider problems in sequential decision making in which uncertainty in the underlying utility function is represented probabilistically [8; 9]. Fishburn [10] also addresses this problem (as we discuss below). Harsanyi’s formulation of games with incomplete information as Bayesian games [12; 13] relies critically on distributions over payoff functions, and virtually the entire literature on in this area adopts this perspective [7; 15].¹

In all of this work, the EEU concept is used to determine the value of decisions in the context of an uncertain utility function. Unfortunately, while EEU has an intuitive appeal, this scheme is sensitive to positive affine transformations of

¹In some sense, much work in collaborative filtering [3; 16] and related models [4] can be viewed as incorporating distributions over utility functions. However, these are used for purposes of classification (i.e., determining a unique utility function for a particular user) and generally uncertainty in utility is not accounted for when making decisions.

the utility functions in question. Implicit in such a scheme is a *commensurability assumption* that allows the quantities present in the different utility functions to be meaningfully compared and combined. This is not always the case. The aim of this paper is to describe certain conditions under which this commensurability assumption can be justified by appeal to the foundational axioms for decision theory as proposed by von Neumann and Morgenstern [19] and Savage [17].

The setting we consider is one in which an agent for a decision maker or user is uncertain about the user's preferences, but wishes to recommend (or take) decisions on the user's behalf. Fishburn [10] has considered the problems of the foundations of expected utility from a somewhat different perspective. He considers the problem in which a decision maker is uncertain about the set of consequences she might face and considers combining utility functions over different consequence sets. Unfortunately, his results cannot be applied (except in a trivial way) to the situation above.²

We begin by defining the problem of decision making given uncertainty over utility functions and the EEU concept. We then examine the sensitivity of EEU to the precise representation of the underlying utility functions, and propose an interpretation of utility uncertainty that allows one to prescribe "canonical" utility function representations under specific circumstances. We conclude with a brief discussion of the implications these considerations have for "practical" elicitation.

2 Expected Expected Utility

We begin by establishing notation and basic background with a quick overview of expected utility and then define the notion of expected expected utility formally.

2.1 Expected Utility

Assume a *decision scenario* consisting of a finite set of possible *decisions* D , a finite set of possible outcomes (or states) S , and a distribution function $p \in \Delta(S)$, for each $d \in D$. The term $p(s)$ denotes the probability of outcome s being realized if the system takes decision d . $p(s)$ can be viewed as a vector $\mathbf{p}$ whose i th component is $p(s_i)$ (given a suitable enumeration of outcomes in S).

A *utility function* $u: S \rightarrow \mathbb{R}$ associates utility $u(s)$ with each outcome s . We will generally view u as a $|S|$ -dimensional vector $\mathbf{u}$ whose i th component u_i is $u(s_i)$.³ The *expected utility* of decision d w.r.t. u is:

$$EU(d, u) = \mathbf{p} \cdot \mathbf{u} = \sum_{i \in S} p(s_i) u_i.$$

The optimal decision d^* w.r.t. u is that with *maximum expected utility (MEU)*.

²While our results are general, it is unclear how profitable it is to model a decision maker's uncertainty about her *own* utility function. It can be argued that such uncertainty should be viewed as "traditional" uncertainty about future outcomes, context, etc. Rather than take a stand on this issue, we simply emphasize that an agent can be genuinely uncertain about a user's utility function, and that our model and results apply (in a practical way) to such a setting.

³If u is represented using some more concise model, $\mathbf{u}$ is simply the vector of parameters required for that model.

It is well known that utility functions are invariant under positive affine transformations. That is, the relative expected utility of any pair of decisions (in any decision scenario) will be unaltered by such a transformation of a utility function. This implies that the optimal decision in any decision scenario is unaffected by such a transformation.

More precisely, von Neumann and Morgenstern equate [19] (classes of) utility functions with preferences over *lotteries*. Let $\langle \frac{1}{2}, s_1; \frac{1}{2}, s_2; \dots, \frac{1}{n}, s_n \rangle$, where $\sum_{i=1}^n \frac{1}{n} = 1$, denote a *simple lottery* over outcomes, with each outcome s_i obtained with probability $\frac{1}{n}$. As a shorthand, we sometimes omit outcomes whose probability is zero. Note that an outcome s_i is itself a (trivial) simple lottery, and that each decision d induces a simple lottery $l(d)$ over outcomes. A *compound lottery* is a lottery whose elements may be further lotteries. Let $\succ$ a preference function over lotteries, with $l_1 \succ l_2$ meaning that l_1 is strictly preferred to l_2 . The relations $\succeq$, $\prec$, $\preceq$, and $\sim$ are defined in the usual way. Assuming certain (relatively uncontroversial) axioms restricting the form of the preference function $\succ$, von Neumann and Morgenstern show that there exists a utility function u that exactly represents $\succ$ in the following sense: $EU(d, u) > EU(d', u)$ iff $l(d) \succ l(d')$. Furthermore, the utility function u is unique up to positive affine transformation. Because of this, we can partition the space of utility functions into equivalence classes, each corresponding to a unique preference ordering $\succ$. We denote this class by $\succ$.

Several of the axioms used to equate utility functions with preferences over lotteries are listed here. These are based on the work of von Neumann and Morgenstern [19] and Savage [17], though the particular form used here is drawn from [14]. We use s^* to denote some most preferred outcome in S (i.e., $s^* \succeq s, \forall s \in S$), and s_* to denote some least preferred outcome (these are guaranteed to exist due to other axioms).

Monotonicity $\langle p, s^*; 1-p, s_* \rangle \succeq \langle q, s^*; 1-q, s_* \rangle$ iff $p \geq q$.

Continuity For each s_i , there is some α such that $s_i \sim \langle \alpha, s^*; 1-\alpha, s_* \rangle$.

Reduction of Compound Lotteries Let l be the lottery $\langle \frac{1}{n}, l_1; \dots; \frac{1}{n}, l_n \rangle$ where each l_i is a lottery of the form $\langle \frac{1}{m}, l_1^i; \dots; \frac{1}{m}, l_m^i \rangle$. Let $l_k, k \leq n$ denote the (unique) lotteries within the set $\{l_j^i \mid i \leq n, j \leq m(i)\}$. Let r be the (reduced) lottery over the (unique) component lotteries l_k with probability $p_k = \sum_{i=1}^n \frac{1}{n} \frac{1}{m(i)} = \frac{1}{m}$ associated with each l_k . Then $r \sim l$.

2.2 Uncertainty over Utility Functions

An agent will often not know the user's utility function u with certainty. We model this uncertainty using a density over the set of utility functions $U \subseteq \mathbb{R}^S$ (or a distribution over a finite support set contained in U). If a system makes a decision d under such conditions of uncertainty, the expected utility of d must reflect this. We consider the following definition for the expected utility of d given density μ over U :

$$EU(d, \mu) = \int \mathbf{p} \cdot \mathbf{u}(\mathbf{u}) d\mu.$$

We refer to this as the *expected expected utility (EEU)* of decision d , since it is the expectation of $EU(d, u)$ w.r.t. $\mu(U)$.

This definition is precisely that used in [5; 6; 1] in the context of utility elicitation, and also that used in much other work involving uncertainty over utility [12; 8; 9]. In such a state of uncertainty—or *belief state*—the optimal decision is that d with maximum EEU $EU(d, \cdot)$. We denote by $EU(\cdot)$ the value of being in belief state $\cdot$, assuming one is forced to make a decision:

$$EU(\cdot) = \max_{d \in D} EU(d, \cdot).$$

We call this generic decision rule the *MEEU decision rule*, by analogy with the classical MEU decision rule.

EEU seems to be a fairly natural concept given probabilistically quantified uncertainty over utilities. The fact that it occurs in many different contexts certainly attests to this fact. Unfortunately, the proposed definition can induce certain anomalies, as we examine below.

3 Justifying MEEU

3.1 Loss of Invariance

The results of von Neumann and Morgenstern suggest that the decisions one makes with respect to belief state $\cdot$ over U should be invariant to legitimate transformations of the elements of U . Certainly, this would be a desirable feature of the MEEU decision rule. One might even claim that the decision rule can only be considered useful if it satisfies this condition. In general, unfortunately, this is not the case.

As a simple illustration, suppose we have a domain with two outcomes s_1 and s_2 , and a distribution $\cdot$ that assigns probability .5 to $u_1 = \langle 1, 3 \rangle$ and probability .5 to $u_2 = \langle 2, 1 \rangle$. Suppose we use the MEEU decision rule in this context, by computing

$$EU(d, \cdot) = \sum_i \sum_j (s_j) u_i(s_j) (u_i).$$

and choosing the decision d with maximum expected utility $EU(d, \cdot)$. Then a decision that accords higher probability to s_2 will be preferred to one that gives lower probability to s_2 . However, if we transform u_2 into $u_2 = \langle 2, 1 \rangle$, the relative utilities of these decisions will be reversed. Thus, the MEEU decision rule is not insensitive to transformations of individual utility functions with positive support. Note that we are not suggesting that agent's will arbitrarily transform some utility functions and not others.⁴ Rather, the question is: which representation of a specific utility function (e.g., u_2 in the example) should be adopted in the first place?

One possible way to deal with this problem is to recognize that a utility function is simply a convenient (and nonunique) way of expressing preferences over lotteries. Rather than working with utility functions, we could work explicitly with a density over preference functions (in fact, we will do this implicitly below). Unfortunately, the set of lotteries over which a preference ordering is defined is uncountable; therefore, some compact representation (of the *individual* preference functions) is needed. But this is precisely the role of

⁴If the same transformation is applied to all functions with positive support, the MEEU decision is unchanged.

a utility function—to serve as a concise representation of a preference function over lotteries.

This gives rise to the question of how to choose a representative utility function from each equivalence class $\succsim$ that allows formal justification of the MEEU decision rule, and under what circumstances such representatives exist.

3.2 A Lottery Interpretation of MEU

We give a formal justification for the MEEU rule under a specific condition: *we assume the existence of a known best and worst outcome*. That is, each utility function with positive support has the same best outcome s and worst outcome s . We also insist that the user is not indifferent to these alternatives, that is, that s must be strictly preferred to s .⁵ We call such utility functions *extremum equivalent*. In many settings, such as those involving active preference elicitation, restricting attention to a set of extremum equivalent utility functions is not problematic. One simply needs to ask the user to identify her most and least preferred outcomes (these need not be unique, but only one such representative need be identified).

Once we know outcomes s and s , we insist that our belief state $\cdot$ assign nonzero measure only to normalized utility functions u in which $u(s) = c$ and $u(s) = c$ for some constants $c > c$. For convenience, we will assume $c = 1$ and $c =$, but nothing crucial depends on these choices. Under these conditions, every preference ordering $\succsim$ for lotteries has a unique utility function representation satisfying the normalization constraints.

We note that each preference ordering corresponds to a unique set of indifference conditions between outcomes and standard lotteries. A *standard lottery* is one of the form $\langle s, s; 1 - \cdot, s \rangle$. By the Continuity Axiom, we have, for each outcome s_i and each preference ordering $\succsim$,

$$s_i \sim \langle \cdot, s; 1 - \cdot, s \rangle,$$

for some unique $\cdot$. We denote by $l(\succsim, s_i)$ this standard lottery. This implies that $u(\cdot(s_i)) = \cdot$ in the normalized utility function u corresponding to $\succsim$.⁶

Now given a belief state $\cdot$, how does one compare the value of different decisions d ? This can be reduced to a question of how one compares the value of guaranteed outcomes s_i , since decisions are just lotteries over outcomes. Given a particular preference ordering $\succsim$, we have $s_i \sim l(\succsim, s_i)$. However, the choice facing the agent for the decision maker involves uncertainty over the true preference ordering. Thus, we can view the outcome s_i as a compound lottery, where first a preference ordering is chosen according to distribution $\cdot$, and then the standard gamble involving s_i is played. This will induce a new preference ordering $\succsim$ over outcomes (and decisions). We illustrate this for the case of a discrete distribution over two preference orderings $\succsim$ and $\succsim$, with probabilities $\cdot$ and $1 - \cdot$ respectively. In this case, the outcome s_i is equated with the compound lottery

$$s_i \sim \langle \cdot, l(\succsim, s_i); 1 - \cdot, l(\succsim, s_i) \rangle.$$

⁵If $s_{\top} \sim s_{\perp}$, then the decision problem is trivial since each decision is equally preferred. Indeed, in some axiomatizations, non-triviality is imposed on legitimate preferences [11].

⁶If the normalizing constants differ from 0/1, then utility is some linear function of $p_i^{\succsim}$. This has no impact on our argument.

By reduction of compound lotteries, this means that

$$s_i \sim_P \langle q \cdot p_i^{\succ} + (1-q) \cdot p_i^{\succ'}, s_{\top}; q \cdot (1-p_i^{\succ}) + (1-q) \cdot (1-p_i^{\succ'}), s_{\perp} \rangle.$$

Similarly, for any other outcome s_j , we have

$$s_j \sim_P \langle q \cdot p_j^{\succ} + (1-q) \cdot p_j^{\succ'}, s_{\top}; q \cdot (1-p_j^{\succ}) + (1-q) \cdot (1-p_j^{\succ'}), s_{\perp} \rangle.$$

By the Monotonicity Axiom, we have $s_i \succ s_j$ iff

$$q \cdot p_i^{\succ} + (1-q) \cdot p_i^{\succ'} > q \cdot p_j^{\succ} + (1-q) \cdot p_j^{\succ'}.$$

Since $p_i^{\succ} = u(s_i)$ (and similarly for the other terms), we have $s_i \succ s_j$ iff $EU(s_i, \succ) > EU(s_j, \succ)$ (where we emphasize that here we are treating outcomes as deterministic decisions that guarantee the corresponding outcomes).⁷ From this one can easily show that for any two decisions d, d' (that induce distributions over outcomes), $d \succ d'$ iff $EU(d, \succ) > EU(d', \succ)$.

This argument applies to arbitrary discrete distributions, and can be generalized to continuous densities as follows:

$$s_i \succ s_j$$

$$\text{iff } l(\succ, s_i) \succ l(\succ, s_j)$$

$$\text{iff } \int p_i(u) \cdot u \, d\mu + (1 - \int p_i(u) \, d\mu) \cdot u_{\top} > \int p_j(u) \cdot u \, d\mu + (1 - \int p_j(u) \, d\mu) \cdot u_{\top}$$

$$\text{iff } \int p_i(u) \cdot u \, d\mu > \int p_j(u) \cdot u \, d\mu$$

$$\text{iff } \int u(s_i) \cdot p_i(u) \, d\mu > \int u(s_j) \cdot p_j(u) \, d\mu$$

$$\text{iff } EU(s_i, \succ) > EU(s_j, \succ)$$

Here the first step refers to a compound lottery over an continuous set of component (simple) lotteries, while we assume in second step that a such a compound lottery can be reduced to a simple lottery in an analogous way to the reduction of a finite compounded lottery.

Thus under the assumption that one can identify a best and worst outcome, the MEEU decision rule can be justified for use with normalized (extremum equivalent) utility functions by appeal to the foundational axioms of decision theory, and an interpretation of uncertainty over utility as a lottery over the lotteries defined by the component utility functions.

We now formalize the legitimacy of EEU and MEE.

Definition 1 Let $\{\succ_i\}$ be a set of extremum equivalent preference relations with respect to finite outcome set S , with best and worst outcomes $s_{\top}$ and $s_{\perp}$, respectively. Let μ be a density over $\{\succ_i\}$. For any $\succ_i$, let $p_i^{\succ}$ denote the tradeoff probability for state s in its standard gamble w.r.t. $\succ_i$; that is,

$$s \sim \langle p_i^{\succ}, s_{\top}; 1 - p_i^{\succ}, s_{\perp} \rangle.$$

⁷Note that if we chose other normalizing constants for our best and worst outcomes, then we would have that $u_{\succ}(s_i)$ is some linear function of $p_i^{\succ}$; but this linear function is identical for each term in the equation, so the conclusion holds.

The *aggregate standard gamble* for $s \in S$ induced by $\{\succ_i\}$ and μ is defined:

$$l = \langle \int p_i^{\succ}(s) \, d\mu, s_{\top}; 1 - \int p_i^{\succ}(s) \, d\mu, s_{\perp} \rangle.$$

Let l be any lottery w.r.t. S . The *aggregate reduction* of l is the standard gamble

$$l^{\text{agg}} = \langle \int p_i^{\succ}(s) \, d\mu, s_{\top}; 1 - \int p_i^{\succ}(s) \, d\mu, s_{\perp} \rangle$$

obtained by replacing every outcome s in l by its aggregate standard gamble and reducing it to a standard gamble in the usual fashion. The *aggregate preference relation* $\succ$ (w.r.t. S) induced by $\{\succ_i\}$ and μ is given by

$$l_1 \succ l_2 \text{ iff } l_1^{\text{agg}} \succ l_2^{\text{agg}}.$$

Theorem 1 Let $\{\succ_i\}$ and μ be defined as above. Let $\{u_i\}$ be a set of utility functions (one for each $\succ_i$, consistent with $\succ_i$) such that $u_i(s_{\top}) = c_{\top}$ and $u_i(s_{\perp}) = c_{\perp}$, for all u_i and two fixed constants $c_{\top} > c_{\perp} \geq 0$. Then the utility function $u(s) = \int u_i(s) \, d\mu$ ($\succ_i$) is consistent with the aggregate preference relation $\succ$ induced by $\{\succ_i\}$ and μ .

Extremum equivalence is thus sufficient to ensure commensurability, as it puts all utility functions on a common scale. It is important to realize that the scale dictated by the best and worst outcomes *cannot vary*, since these are truly best and worst outcomes; we return to this point below. It appears to be much more difficult to apply this type of argument to densities over utility functions that are not extremum equivalent.

Fishburn [10] considers the problem of EEU when a decision maker is uncertain about the nature of the consequence sets she will face. He proposes foundational axioms that justify the use EEU to compare decisions. However, the setting is rather different: specifically, Fishburn requires that any consequences that two utility functions have in common be ranked identically. In our context, where each utility function lies over the same consequence set, the Fishburn axioms impose overly stringent requirements. It is interesting to note that Fishburn requires something akin to extremum equivalence, namely, that there exist two consequences common to the domains of each utility function such that one of the consequences is preferred to the other in each function.

4 Dealing with Small Worlds

It is important to realize that the best and worst outcomes with which one calibrates must either be truly best and worst outcomes from the decision maker's standpoint, or they themselves must be calibrated. Using Savage's [17] terminology, we must be careful to distinguish "small worlds" reasoning from "grand worlds." Consider the case where the set of outcomes is restricted to the subset of outcomes that are possible given the set of actions in a specific decision scenario. But assume there exist outcomes outside the domain of the restricted scenario for which the user has concrete preferences. Let's refer to the set of restricted outcomes as *local*, while the space of all outcomes is *global*.

We might imagine determining the best and worst *local* outcomes, s^l and s^l , respectively, and engaging in the elicitation process using standard gambles with respect to these extreme local outcomes. Unfortunately, this is not sufficient to justify the MEEU decision rule. The difficulty is that the user's degrees of preference for s^l and s^l may themselves be characterized by uncertainty. Specifically, s^l can be equated with a standard gamble $\langle \frac{1}{i}, s; 1 - \frac{1}{i}, s^l \rangle$ (where s, s^l denote the globally best and worst outcomes). Similarly, $s^l \sim \langle \frac{1}{j}, s; 1 - \frac{1}{j}, s^l \rangle$. But the relevant gamble probabilities— $\frac{1}{i}$ and $\frac{1}{j}$ —may vary with the utility function and will not be known if attention is restricted to the *small worlds* domain. As a consequence, our distribution over *local* utility functions is insufficient to justify MEEU.

To illustrate, suppose that we believe the user's utility function is either u_1 (with probability $\frac{1}{2}$) or u_2 (with probability $\frac{1}{2} = 1 - \frac{1}{2}$). Furthermore, suppose that we consider only the projection of these onto the set of local outcomes, calibrating two outcomes s_i and s_j with respect to "local" standard gambles. Suppose that we have the following tradeoffs with respect to the local extrema in u_1 :

$$s_i \sim_1 \langle \frac{1}{i}, s^l; (1 - \frac{1}{i}), s^l \rangle$$

$$s_j \sim_1 \langle \frac{1}{j}, s^l; (1 - \frac{1}{j}), s^l \rangle$$

and similarly in u_2 :

$$s_i \sim_2 \langle \frac{2}{i}, s^l; (1 - \frac{2}{i}), s^l \rangle$$

$$s_j \sim_2 \langle \frac{2}{j}, s^l; (1 - \frac{2}{j}), s^l \rangle.$$

Using EEU, we have

$$s_i \sim \langle \frac{1}{2} \frac{1}{i} + \frac{1}{2} \frac{2}{i}, s^l; \frac{1}{2} (1 - \frac{1}{i}) + \frac{1}{2} (1 - \frac{2}{i}), s^l \rangle$$

$$s_j \sim \langle \frac{1}{2} \frac{1}{j} + \frac{1}{2} \frac{2}{j}, s^l; \frac{1}{2} (1 - \frac{1}{j}) + \frac{1}{2} (1 - \frac{2}{j}), s^l \rangle.$$

Now, were we to place this *small world* in the *grand world* context, we might say that the local best and worst outcomes, s^l and s^l , have specific lottery probabilities with respect to the global utility functions. Let's assume that we have

$$s^l \sim_1 \langle \frac{1}{i}, s; (1 - \frac{1}{i}), s \rangle$$

$$s^l \sim_1 \langle \frac{1}{j}, s; (1 - \frac{1}{j}), s \rangle$$

$$s^l \sim_2 \langle \frac{2}{i}, s; (1 - \frac{2}{i}), s \rangle$$

$$s^l \sim_2 \langle \frac{2}{j}, s; (1 - \frac{2}{j}), s \rangle.$$

In other words, in (the normalized, global counterpart of) utility function u_1 , we have $u_1(s^l) = \frac{1}{i}$ and $u_1(s^l) = \frac{1}{j}$; while in u_2 we have $u_2(s^l) = \frac{2}{i}$ and $u_2(s^l) = \frac{2}{j}$. Applying EEU and reduction of compound lotteries, we see that

$$\begin{aligned} s_i \sim & \langle \frac{1}{2} \frac{1}{i} + \frac{1}{2} \frac{2}{i} + \frac{1}{2} \frac{1}{i} (1 - \frac{1}{i}) + \frac{1}{2} \frac{2}{i} (1 - \frac{2}{i}), s; \\ & (1 - \frac{1}{i}) \frac{1}{i} + (1 - \frac{2}{i}) \frac{2}{i} \\ & + (1 - \frac{1}{i}) \frac{1}{i} (1 - \frac{1}{i}) + (1 - \frac{2}{i}) \frac{2}{i} (1 - \frac{2}{i}), s \rangle \end{aligned}$$

and

$$\begin{aligned} s_j \sim & \langle \frac{1}{2} \frac{1}{j} + \frac{1}{2} \frac{2}{j} + \frac{1}{2} \frac{1}{j} (1 - \frac{1}{j}) + \frac{1}{2} \frac{2}{j} (1 - \frac{2}{j}), s; \\ & (1 - \frac{1}{j}) \frac{1}{j} + (1 - \frac{2}{j}) \frac{2}{j} \\ & + (1 - \frac{1}{j}) \frac{1}{j} (1 - \frac{1}{j}) + (1 - \frac{2}{j}) \frac{2}{j} (1 - \frac{2}{j}), s \rangle. \end{aligned}$$

Hence, $s_i \succ s_j$ iff

$$\begin{aligned} & \frac{1}{2} \frac{1}{i} + \frac{1}{2} \frac{2}{i} + \frac{1}{2} \frac{1}{i} (1 - \frac{1}{i}) + \frac{1}{2} \frac{2}{i} (1 - \frac{2}{i}) \\ & > \frac{1}{2} \frac{1}{j} + \frac{1}{2} \frac{2}{j} + \frac{1}{2} \frac{1}{j} (1 - \frac{1}{j}) + \frac{1}{2} \frac{2}{j} (1 - \frac{2}{j}). \end{aligned}$$

Unfortunately, this condition is not equivalent to the MEEU rule in the original small worlds domain. Specifically, this condition cannot generally be assessed without having some assessment of the global tradeoff probabilities $\frac{1}{i}$, etc. In other words, to accurately compare two small world outcomes given uncertainty about the *local* utility functions, we have to explicitly assess our uncertainty about the *range* of values the local extrema can take with respect to the *global* utility function. Thus while one can generally use small world reasoning in the classic decision-theoretic setting, its use is problematic in the EEU framework.

Fortunately, the MEEU principle can be recovered if we make one simple assumption: that the global tradeoffs associated with the best and worst local outcomes do not vary with the utility function (at least, with those utility functions having positive support). For example, to continue with the illustration above, suppose that the tradeoff probabilities associated with the two utilities u_1 and u_2 are identical; that is, $\frac{1}{i} = \frac{2}{i}$ and $\frac{1}{j} = \frac{2}{j}$. (Whether the probabilities are known is irrelevant, all that matters is that they are known to be identical.) If this is so we have $s_i \succ s_j$ iff

$$\begin{aligned} & \frac{1}{2} (\frac{1}{i} + \frac{2}{i}) + \frac{1}{2} (\frac{1}{i} (1 - \frac{1}{i}) + \frac{2}{i} (1 - \frac{2}{i})) \\ & > \frac{1}{2} (\frac{1}{j} + \frac{2}{j}) + \frac{1}{2} (\frac{1}{j} (1 - \frac{1}{j}) + \frac{2}{j} (1 - \frac{2}{j})) \end{aligned}$$

Since the inner terms in each expression sum to one, and $\frac{1}{i} > \frac{1}{j}$ (since $s^l \succ s^l$), we must have $s_i \succ s_j$ iff

$$\frac{1}{i} + \frac{2}{i} > \frac{1}{j} + \frac{2}{j}$$

This, of course, is the expression for MEEU.

Indeed, we don't have to assume that the global tradeoff probabilities are fixed. It is sufficient to assume that the unknown range of (global) utility spanned by s^l and s^l is probabilistically independent of the unknown local utility function u . This too is sufficient to allow MEEU to be used justifiably. We omit the argument, which is straightforward.

5 Consequences for Preference Elicitation

The considerations above have implications for practical preference elicitation. From a foundational perspective, calibration of utilities relative to known best and worst outcomes is required if decisions are to be based on EEU. In the case of incremental elicitation, where EEU is used to determine value of information, we must first obtain a prior over a set of extremum equivalent utility functions before engaging in such deliberations. Fortunately, it often seems to make sense to determine best and worst outcomes beforehand, and engage in "serious" elicitation after this initial calibration.

Another important question: how does one determine priors over utility functions? Utility function databases [4] could be used. This poses some problems regarding interpersonal utility comparison for which there are no especially compelling solutions. When using EEU, things are a bit worse: we need to construct priors conditioned on the observed or

elicited best and worst outcomes. Given a prior over arbitrary utility functions, as long as decisions using EEU are not made until the determination of best and worst outcomes is completed, this poses no difficulties. An alternative, in certain scenarios, is to suppose that certain outcomes are universally most and least preferred (e.g., in medical contexts, death can be used as the latter). This may be hard to justify formally, but from a practical point of view will be quite useful and (one hopes) have little impact on actual decision quality.

The issue of small worlds also poses certain problems. From the point of view of practical elicitation, the prospect of calibrating some small set of relevant outcomes using a user's "global" utility function is unappealing. Fortunately, as argued above, for a given individual, strength of preference can be assumed fixed for the best and worst outcomes, which allows things to carry through. Strength of preference may prove to be important however when trading off the increase in EEU with the effort associated with the elicitation process. Furthermore, this can have an important impact on the construction of priors from databases of utility functions.

6 Concluding Remarks

Decision making when the underlying utility function is unknown is an important problem in game theory, interactive optimization, and preference elicitation. Quantifying this uncertainty using distributions over utility functions has a number of appealing qualities, and quite naturally leads to the notion of *expected* expected utility, a concept that has been used in several different lines of research.

The aim of this paper is to point out that expectations taken with respect to utility function distributions require some care. More precisely, the operation of expected expected utility only makes sense (from a foundational standpoint) when the distribution is over extremal equivalent utility functions. While this has certain implications for practical utility elicitation, we have argued that this requirement is not overly stringent from a practical perspective.

Acknowledgements

Thanks to Radford Neal whose comments on preference elicitation led to this investigation and to Moshe Tennenholtz for suggesting interesting connections. This work was supported by the Institute for Robotics and Intelligent Systems (IRIS), and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

- [1] Craig Boutilier. A POMDP formulation of preference elicitation problems. In *Proc. Eighteenth National Conference on Artificial Intelligence*, pages 239–246, Edmonton, 2002.
- [2] Craig Boutilier, Fahiem Bacchus, and Ronen I. Brafman. UCP-Networks: A directed graphical representation of conditional utilities. In *Proc. Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 56–64, Seattle, 2001.
- [3] Jack S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, Madison, WI, 1998.
- [4] U. Chajewska, L. Getoor, J. Norman, and Y. Shahar. Utility elicitation as a classification problem. In *Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 79–88, Madison, WI, 1998.
- [5] Urszula Chajewska and Daphne Koller. Utilities as random variables: Density estimation and structure discovery. In *Proc. Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 63–71, Stanford, 2000.
- [6] Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *Proc. Seventeenth National Conference on Artificial Intelligence*, pages 363–369, Austin, TX, 2000.
- [7] Tommy Chin-Chiu Chan and Sérgio Ribeiro da Costa Werlang. The Bayesian foundations of the solution concepts of games. *Journal of Economic Theory*, 45:370–391, 1988.
- [8] Richard M. Cyert and Morris H. de Groot. Adaptive utility. In Maurice Allais and Ole Hagen, editors, *Expected Utility Hypothesis and the Allais Paradox*, pages 223–241. D. Reidel, Dordrecht, 1979.
- [9] Morris H. de Groot. Decision making with an uncertain utility function. In Bernt P. Stigum and Fred Wenstøp, editors, *Foundations of Utility and Risk Theory with Applications*, pages 371–384. D. Reidel, Dordrecht, 1983.
- [10] Peter C. Fishburn. *The Foundations of Expected Utility*. D. Reidel, Dordrecht, 1982.
- [11] Simon French. *Decision Theory*. Halsted Press, New York, 1986.
- [12] John C. Harsanyi. Games with incomplete information played by Bayesian players. part I: The basic model. *Management Science*, 14:159–182, 1967.
- [13] John C. Harsanyi. Games with incomplete information played by Bayesian players. part II: Bayesian equilibrium points. *Management Science*, 14:320–334, 1968.
- [14] R. D. Luce and H. Raiffa. *Games and Decisions*. Wiley, New York, 1957.
- [15] Roger B. Myerson. *Game Theory: Analysis of Conflict*. Harvard University Press, Cambridge, 1991.
- [16] David M. Pennock and Eric Horvitz. Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In *Proc. Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 473–480, Stanford, 2000.
- [17] Leonard J. Savage. *The Foundations of Statistics*. Wiley, New York, 1954.
- [18] Edward A. Sykes and Chelsea C. White, III. Multiobjective intelligent computer-aided design. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6):1498–1511, 1991.
- [19] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, 1944.
- [20] Tianhan Wang and Craig Boutilier. Incremental utility elicitation with the minimax regret decision criterion. In *Proc. Eighteenth International Joint Conference on Artificial Intelligence*, this volume, Acapulco, 2003.
- [21] Chelsea C. White, III, Andrew P. Sage, and Shigeru Dozono. A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE Transactions on Systems, Man and Cybernetics*, 14(2):223–229, 1984.

Great Expectations.

Part I: On the Customizability of Generalized Expected Utility*

Francis C. Chu and Joseph Y. Halpern

Department of Computer Science

Cornell University

Ithaca, NY 14853, U.S.A.

Email: {fcc,halpern}@cs.cornell.edu

Abstract

We propose a generalization of expected utility that we call *generalized EU* (GEU), where a decision maker's beliefs are represented by plausibility measures and the decision maker's tastes are represented by general (i.e., not necessarily real-valued) utility functions. We show that every agent, "rational" or not, can be modeled as a GEU maximizer. We then show that we can customize GEU by selectively imposing just the constraints we want. In particular, by show how of Savage's postulates corresponds to constraints on GEU.

1 Introduction

Many decision rules have been proposed in the literature. Perhaps the best-known approach is based on maximizing expected utility (EU). This approach comes in a number of variants; the two most famous are due to von Neumann and Morgenstern [1947] and Savage [1954]. They all follow the same pattern: they formalize the set of alternatives among which the decision maker (DM) must choose (typically as *acts*¹ or *lotteries*²). They then give a set of assumptions (often called *postulates* or *axioms*) such that the DM's preferences on the alternatives satisfy these assumptions iff the preferences have an EU representation, where an *EU representation* of a preference relation is basically a utility function (and a probability measure when acts are involved) such that the relation among the alternatives based on expected utility agrees with the preference relation. Moreover, they show that the representation is essentially unique. In other words, if the preferences of a DM satisfy the assumptions, then she is behaving *as if* she has quantified her tastes via a real-valued utility function (and her beliefs via a probability measure)

*Work supported in part by NSF under grants IIS-0090145 and CTC-0208535 and by the DoD Multidisciplinary University Research Initiative (MURI) program administered by ONR under grant N00014-01-1-0795.

¹Formally, given a set S of states of the world and another set C of consequences, an *act* a is a function from S to C that, intuitively, associates with each state s the consequence of performing a in s .

²Formally, a *lottery* is a probability distribution over consequences; intuitively, the distribution quantifies how likely it is that each consequence occurs.

and she is relating the alternatives according to their expected utility. The assumptions are typically regarded as criteria for rational behavior, so these results also suggest that if a DM's beliefs are actually described by a probability measure and her tastes are described by a utility function, then she should relate the alternatives according to their expected utility (if she wishes to appear rational).

Despite the appeal of EU maximization, it is well known that people do not follow its tenets in general [Resnik, 1987]. As a result, a host of extensions of EU have been proposed that accommodate some of the more systematic violations (see, for example, [Gul, 1991; Gilboa and Schmeidler, 1989; Giang and Shenoy, 2001; Quiggin, 1993; Schmeidler, 1989; Yaari, 1987]). Again, the typical approach in the decision theory literature has been to prove representation theorems, showing that, given a suggested decision rule $\mathcal{R}$, there exists a collection of assumptions such that a preference relation satisfies the assumptions iff it has an $\mathcal{R}$ representation, where an $\mathcal{R}$ *representation* of a preference relation is essentially a choice of tastes (and beliefs) such that, given these as inputs, $\mathcal{R}$ relates the alternatives the way the preference relation does.

Given this plethora of rules, it would be useful to have a general framework in which to study decision making. The framework should also let us understand the relationship between various decision rules. We provide such a framework in this paper.

The basic idea of our approach is to generalize the notion of expected utility so that it applies in as general a context as possible. To this end, we introduce *expectation domains*, which are structures consisting of

- three (component) domains: a *plausibility domain*, a *utility domain*, and a *valuation domain*,
- two binary operators $\oplus$ and $\otimes$, which are the analogues of $+$ and $\times$ over the reals, and
- a reflexive binary relation $\preceq$ (which generalizes $\leq$).

Intuitively, $\otimes$ combines plausibility values and utility values much the same way that $\times$ combines probability and (real) utility, while $\oplus$ combines the products to form the (generalized) expected utility.

We have three domains because we do not want to require that DMs be able to add or multiply plausibility values or utility values, since these could be qualitative (e.g., plausibility

values could be {"unlikely", "likely", "very likely", ...} and utility values could be {"bad", "good", "better", ...}. In general, we do not assume that $\preceq$ is an order (or even a pre-order), since we would like to be able to represent as many preference relations and decision rules as possible.

Once we have an expectation domain, DMs can express their tastes and beliefs using components of the expectation domain. More specifically, the DMs express their beliefs using a *plausibility measure* [Friedman and Halpern, 1995], whose range is the plausibility domain of the expectation domain (plausibility measures generalize probability measures and a host of other representations of uncertainty, such as sets of probability measures, Choquet capacities, possibility measures, ranking functions, etc.) and they express their tastes using a utility function whose range is the utility domain of the expectation domain. In an expectation domain, it is possible to define a generalization of expected utility, which we call *generalized EU* (GEU). The GEU of an act is basically the sum (in the sense of $\oplus$) of products (in the sense of $\otimes$) of plausibility values and utility values that generalizes the standard definition of (probabilistic) expected utility over the reals in the obvious way.

We start by proving an analogue of Savage's result with respect to the decision rule (Maximizing) GEU.³ We show that every preference relations on acts has a GEU representation (even those that do not satisfy any of Savage's postulates), where a *GEU representation* of a preference relation basically consists of an expectation domain E , plausibility measure P_1 , and utility function u , such that the way acts are related according to their GEU agrees with the preference relation (Theorem 3.1). In other words, no matter what the DM's preference relation on acts is, she behaves as if she has quantified her beliefs by a plausibility measure and her tastes via a utility function, and is relating the acts according to their (generalized) expected utility as defined by the $\oplus$ and $\otimes$ of some expectation domain. That is, we can model *any* DM using GEU, whether or not the DM satisfies any rationality assumptions. An important difference between our result and that of Savage is that he was constructing *EU* representations, which consists of a *real-valued* utility function u and a *probability* measure Pr (and the expectation domain is fixed, so $\oplus$, $\otimes$, and $\preceq$ are just $+$, $\times$, and $\leq$, respectively).

Given that GEU can represent *all* preference relations, it might be argued that GEU is too general—it offers no guidelines as to how to make decisions. We view this as a feature, not a bug, since our goal is to provide a general framework in which to express and study decision rules, instead of proposing yet another decision rule. Thus the absence of "guidelines" is in fact an absence of *limitations*: we do not want to exclude any possibilities at the outset, even preference relations that are not transitive or are incomplete. Starting from

³Many decision rules involve optimizing (i.e., maximizing or minimizing) some value function on the acts. Sometimes it is explicitly mentioned whether the function is to be maximized or minimized (e.g., "Minimax Regret" says explicitly to "minimize the maximum regret") while other times only the function name is mentioned and it is implicitly understood what is meant (e.g., "EU" means "maximize EU"). In this paper we will use "Maximizing GEU" and "GEU" interchangeably.

a framework in which we can represent all preference relations, we can then consider what preference relations have "special" representations, in the sense that the expectation domain, plausibility measure, and utility function in the representation satisfy some (joint) properties. This allows us to show how properties of expectation domains correspond to properties of preference relations. We can then "customize" GEU by placing just the constraints we want. We illustrate this by showing how each of Savage's postulates corresponds in a precise sense to an axiom on GEU.

Intuitively, a decision rule maps tastes (and beliefs) to preference relations on acts. Given two decision rules $\mathcal{R}_1$ and $\mathcal{R}_2$, an $\mathcal{R}_1$ *representation* of $\mathcal{R}_2$ is basically a function τ that maps inputs of $\mathcal{R}_2$ to inputs of $\mathcal{R}_1$ that represent the same tastes and beliefs, with the property that $\mathcal{R}_1(\tau(x)) = \mathcal{R}_2(x)$. Thus, τ models, in a precise sense, a user of $\mathcal{R}_2$ as a user of $\mathcal{R}_1$, since τ preserves tastes (and beliefs). In a companion paper [Chu and Halpern, 2003] we show that (almost) every decision rule has a GEU representation.

Although there has been a great deal of work on decision rules, there has been relatively little work on finding general frameworks for representing decision rules. In particular, there has been no attempt to find a decision rule that can represent all preference relations. There has been work in the fuzzy logic community on finding general notions of integration (which essentially amounts to finding notions of expectation) using generalized notions of $\oplus$ and $\otimes$; see, for example, [Benvenuti and Mesiar, 2000]. However, the expectation domain used in this work is (a subset of) the reals; arbitrary expectation domains are not considered. Luce [1990; 2000] also considers general addition-like operations applied to utilities, but his goal is to model joint receipts (which are typically modeled as commodity bundles in economics) as a binary operation, rather than to represent decision rules.

The rest of this paper is organized as follows. We cover some basic definitions in Section 2: plausibility domains, utility domains, expectation domains, decision problems, and GEU. We show that every preference relation on acts has a GEU representation in Section 3. In Section 4, we show that each of Savage's postulates corresponds to an axiom on GEU. We conclude in Section 5. Proofs of theorems stated are available at <http://www.cs.cornell.edu/home/halpern>.

2 Preliminaries

2.1 Plausibility, Utility, and Expectation Domains

Since one of the goals of this paper is to provide a general framework for all of decision theory, we want to represent the tastes and beliefs of the DMs in as general a framework as possible. In particular, we do not want to force the DMs to linearly preorder all consequences and all events (i.e., subsets of the set of states). To this end, we use plausibility measures to represent the beliefs of the DMs and (generalized) utility functions to represent their tastes.

A *plausibility domain* is a set P , partially ordered by $\preceq_P$ (so $\preceq_P$ is a reflexive, antisymmetric, and transitive relation), with two special elements $\perp_P$ and $\top_P$, such that for all $x \in P$, $\perp_P \preceq_P x \preceq_P \top_P$. Given a set S , a function $P_1 : 2^S \rightarrow P$ is a *plausibility measure* iff

- Pl1. $\text{Pl}(\emptyset) = \perp$,
- Pl2. $\text{Pl}(S) = \top$, and
- Pl3. if $X \subseteq Y$ then $\text{Pl}(X) \preceq \text{Pl}(Y)$.

Clearly plausibility measures are generalizations of probability measures. As pointed out in [Friedman and Halpern, 1995], plausibility measures generalize a host of other representations of uncertainty as well. Note that while the probability of any two sets must be comparable (since $\mathbb{R}$ is totally ordered), the plausibility of two sets may be incomparable.

We also want to represent the tastes of DMs using something more general than $\mathbb{R}$, so we allow the range of utility functions to be utility domains, where a *utility domain* is a set U endowed with a reflexive binary relation $\preceq_U$. Intuitively, elements of U represent the strength of likes and dislikes of the DM while elements of P represent the strength of her beliefs. Note that we do not require the DM's preference to be transitive (although we can certainly add this requirement). Experimental evidence shows that DMs' preferences occasionally do seem to violate transitivity.

Once we have plausibility and utility, we want to combine them to form expected utility. To do this, we introduce expectation domains, which have utility domains, plausibility domains, and operators $\oplus$ (the analogue of $+$) and $\otimes$ (the analogue of $\times$).⁴ More formally, an *expectation domain* is a tuple $E = (U, P, V, \otimes, \oplus)$, where $(U, \preceq_U)$ is a utility domain, $(P, \preceq_P)$ is a plausibility domain, $(V, \preceq_V)$ is a valuation domain (where $\preceq_V$ is a reflexive binary relation), $\otimes : P \times U \rightarrow V$, and $\oplus : V \times V \rightarrow V$. (As usual, we omit subscripts when they are clear.) We have four requirements on expectation domains:

- E1. $(x \oplus y) \oplus z = x \oplus (y \oplus z)$;
- E2. $x \oplus y = y \oplus x$;
- E3. $\top \otimes x = x$;
- E4. $(U, \preceq_U)$ is a substructure of $(V, \preceq_V)$.

E1 and E2 say that $\oplus$ is associative and commutative. E3 says that $\top$ is the left-identity of $\otimes$ and E4 ensures that the expectation domain respects the relation on utility values.

Note that we do not require that $\oplus$ be monotonic; that is, we do not require that for all $x, y, z \in V$,

$$\text{if } x \preceq_V y \text{ then } x \oplus z \preceq_V y \oplus z. \quad (2.1)$$

We say that E is *monotonic* iff (2.1) holds. It turns out that monotonicity does not really make a difference by itself; see the comments after the proof of Theorem 3.1.

2.2 Decision Situations and Decision Problems

A *decision situation* describes the objective part of the circumstance that the DM faces (i.e., the part that is independent of the tastes and beliefs of the DM). Formally, a decision situation is a tuple $\mathcal{A} = (A, S, C)$, where

- S is the set of states of the world,
- C is the set of consequences, and
- A is a set of acts (i.e., a set of functions from S to C).

⁴Sometimes we use $\times$ to denote Cartesian product; the context will always make it clear whether this is the case.

An act a is *simple* iff its range is finite. That is, a is simple if it has only finitely many consequences. Many works in the literature focus on simple acts (e.g., [Fishburn, 1987]). We assume in this paper that A contains only simple acts; this means that we can define (generalized) expectation using finite sums, so we do not have to introduce infinite series or integration for arbitrary expectation domains. Note that all acts are guaranteed to be simple if either S or C is finite, although we do not assume that here.

A decision problem is essentially a decision situation together with information about the tastes and beliefs of the DM; that is, a decision problem is a decision situation together with the subjective part of the circumstance that faces the DM. Formally, a (*plausibilistic*) *decision problem* is a tuple $\mathcal{D} = (\mathcal{A}, E, \mathbf{u}, \text{Pl})$, where

- $\mathcal{A} = (A, S, C)$ is a decision situation,
- $E = (U, P, V, \otimes, \oplus)$ is an expectation domain,
- $\mathbf{u} : C \rightarrow U$ is a utility function, and
- $\text{Pl} : 2^S \rightarrow P$ is a plausibility measure.

We say that $\mathcal{D}$ is *monotonic* iff E is monotonic.

2.3 Expected Utility

Let $\mathcal{D} = ((A, S, C), E, \mathbf{u}, \text{Pl})$ be a plausibilistic decision problem. Each $a \in A$ induces a *utility random variable* $\mathbf{u}_a : S \rightarrow U$ as follows: $\mathbf{u}_a(s) = \mathbf{u}(a(s))$. In the standard setting (where utilities are real-valued and Pl is a probability measure Pr), we can identify the expected utility of act a with the expected value of $\mathbf{u}_a$ with respect to Pr , computed in the standard way:

$$\mathbf{E}_{\text{Pr}}(\mathbf{u}_a) = \sum_{x \in \text{ran}(\mathbf{u}_a)} \text{Pr}(\mathbf{u}_a^{-1}(x)) \times x. \quad (2.2)$$

We can generalize (2.2) to an arbitrary expectation domain $E = (U, P, V, \otimes, \oplus)$ by replacing $+$, $\times$, and Pr by $\oplus$, $\otimes$, and Pl , respectively. This gives us

$$\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a) = \bigoplus_{x \in \text{ran}(\mathbf{u}_a)} \text{Pl}(\mathbf{u}_a^{-1}(x)) \otimes x. \quad (2.3)$$

We call (2.3) the *generalized EU* (GEU) of act a . Clearly (2.2) is a special case of (2.3).

In the probabilistic case, if all singleton sets are measurable with respect to Pr (i.e., in the domain of Pr), then

$$\mathbf{E}_{\text{Pr}}(\mathbf{u}_a) = \sum_{s \in S} \text{Pr}(s) \times \mathbf{u}_a(s). \quad (2.4)$$

The plausibilistic analogue of (2.4) is not necessarily equivalent to (2.3). A decision problem $((A, S, C), E, \mathbf{u}, \text{Pl})$ is *additive* iff

$$\text{Pl}(X \cup Y) \otimes \mathbf{u}(c) = \text{Pl}(X) \otimes \mathbf{u}(c) \oplus \text{Pl}(Y) \otimes \mathbf{u}(c)$$

for all $c \in C$ and nonempty $X, Y \subseteq S$ such that $X \cap Y = \emptyset$. Note that the notion of additivity we defined is a joint property of several components of a decision problem (i.e., $\oplus$, $\otimes$, $\mathbf{u}$, and Pl) instead of being a property of Pl alone. Additivity is exactly the requirement needed to make the analogue of (2.4) equivalent to (2.3). While decision problems involving probability are additive, those involving representations of uncertainty such as Dempster-Shafer belief functions are not, in general.

3 Representing Arbitrary Preference Relations

In this section, we show that every preference relation on acts has a GEU representation. GEU, like all decision rules, is formally a function from decision problems to preference relations on acts. Thus a GEU representation of a preference relation $\lesssim_A$ on the acts in $\mathcal{A} = (A, \dots)$ is a decision problem $\mathcal{D} = (\mathcal{A}, E, \mathbf{u}, \text{Pl})$, where $E = (U, P, V, \oplus, \otimes)$, such that $a_1 \lesssim_A a_2$ iff $\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_{a_1}) \lesssim_V \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_{a_2})$.

Theorem 3.1: Every $\lesssim_A$ has a GEU representation.⁵

Proof: Fix some $\mathcal{A} = (A, S, C)$ and $\lesssim_A$. We want to construct a decision problem $\mathcal{D} = (\mathcal{A}, E, \mathbf{u}, \text{Pl})$ such that $\text{GEU}(\mathcal{D}) = \lesssim_A$.

The idea is to let each consequence be its own utility and each set be its own plausibility, and define $\otimes$ and $\oplus$ such that each act is its own expected utility. For each $c \in C$, let a_c denote the constant act with the property that $a_c(s) = c$ for all $s \in S$. Let $E = (U, P, V, \oplus, \otimes)$ be defined as follows:

1. $U = (C, \lesssim_C)$, where $c \lesssim_C d$ iff $c = d$ or $a_c, a_d \in A$ and $a_c \lesssim_A a_d$. (Note that Savage assumes that A contains all simple acts; in particular, A contains all constant acts. We do not assume that here.)
2. $P = (2^S, \subseteq)$.
3. $V = (2^{S \times C}, \lesssim_V)$, where $x \lesssim_V y$ iff $x = y$ or $x, y \in A$ and $x \lesssim_A y$. (Note that set-theoretically a function is a set of ordered pairs, so $A \subseteq 2^{S \times C}$.)
4. $x \oplus y = x \cup y$ for $x, y \in V$.
5. $X \otimes c = X \times \{c\}$ for $X \in 2^S (= P)$ and $c \in C (= U)$.

We can identify $c \in C$ with $S \times \{c\}$ in V ; with this identification, $(U, \lesssim_U)$ is a substructure of $(V, \lesssim_V)$ and $\top \otimes c = c$ for all $c \in U (= C)$, as required. Furthermore, $\oplus$ is clearly associative and commutative, so E is indeed an expectation domain. Let $\mathcal{D} = (\mathcal{A}, E, \mathbf{u}, \text{Pl})$, where $\mathbf{u}(c) = c$ and $\text{Pl}(X) = X$. Note that

$$\begin{aligned} \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a) &= \bigoplus_{x \in \text{ran}(\mathbf{u}_a)} \text{Pl}(\mathbf{u}_a^{-1}(x)) \otimes x \\ &= \bigoplus_{c \in \text{ran}(a)} \text{Pl}(a^{-1}(c)) \otimes c \\ &= \{(s, c) \mid a(s) = c\} \\ &= a. \end{aligned}$$

That is, each act is its own expected utility; by the definition of $\lesssim_V$, it is clear that $a \lesssim_A b$ iff $\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a) \lesssim_V \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_b)$. Thus $\text{GEU}(\mathcal{D}) = \lesssim_A$, as desired. ■

Note that the representation constructed in Theorem 3.1 is in fact additive, since if $X \cap Y = \emptyset$, then

$$(X \cup Y) \times \{c\} = (X \times \{c\}) \cup (Y \times \{c\}).$$

The representation is not necessarily monotonic. However, as we show in the full paper, a simple modification to the definition of $\lesssim_V$ makes the representation monotonic, so monotonicity by itself does not restrict the kind of preference relations that can be represented.

⁵Note that, unlike most representation theorems, there is no uniqueness condition. However, we can show that our representation is canonical in a certain sense.

Theorem 3.1 holds in large part because of the flexibility we have. Given a decision situation (A, S, C) and a preference relation $\lesssim_A$ on A , we are able to construct an expectation domain and a relation $\lesssim_V$ that is customized to capture the relation $\lesssim_A$ on A . However, this depends (in part) on two features of our setup.

The first is that, following Savage [1954], we took acts to be functions from states to consequences. Suppose instead that we have a *consequence function* $c : A \times S \rightarrow C$ that takes an act a and a state s and gives the consequence of a in s . Of course, in this setting, two distinct acts a_1 and a_2 could induce the same function from states to consequences; that is, we might have $c(a_1, s) = c(a_2, s)$ for all $s \in S$. It is easy to see that if a_1 and a_2 induce the same function from states to consequences, then no matter what expectation domain, utility function, and plausibility measure we use, a_1 and a_2 will have the same expected utility. Thus, if $\lesssim_A$ does not treat a_1 and a_2 the same way, then $\lesssim_A$ has no GEU representation.

A second reason that we do not need consistency constraints on $\lesssim_A$ is that we have placed no constraints on $\lesssim_V$, and relatively few constraints on $\oplus$, $\otimes$, $\mathbf{u}$, and Pl . If, for example, we required $\lesssim_V$ to be transitive, then we would also have to require that $\lesssim_A$ be transitive. The lack of constraints on $\oplus$, $\otimes$, $\mathbf{u}$, and Pl is important because it gives us enough freedom to ensure that distinct acts have different expected utility. In the next section, we investigate what happens when we add more constraints.

4 Representing Savage's Postulates

Theorem 3.1 shows that GEU can represent any preference relation. We are typically interested in representing preference relations that satisfy certain constraints, or postulates. The goal of this section is to examine the effect of such constraints on the components that make up GEU. For definiteness, we focus on Savage's postulates. For ease of exposition, we restrict to additive decision problems in this section; recall that this restriction does not affect Theorem 3.1.

A set $\mathcal{P}_e$ of axioms about (i.e., constraints on) plausibilistic decision problems *represents* a set of postulates $\mathcal{P}_r$ about decision situations and preference relations iff

$$\mathcal{D} = (\mathcal{A}, \dots) \text{ satisfies } \mathcal{P}_e \text{ iff } (\mathcal{A}, \text{GEU}(\mathcal{D})) \text{ satisfies } \mathcal{P}_r.$$

Theorem 3.1 can be viewed as saying that the empty set of axioms represents the empty set of postulates.

Before we present Savage's postulates, we first introduce some notation that will make the exposition more succinct. Suppose that $f : X \rightarrow Y$, $g : X \rightarrow Y$, and $Z \subseteq X$. Let $\langle f, Z, g \rangle$ denote the function h such that $h(x) = f(x)$ for all $x \in Z$ and $h(x) = g(x)$ for all $x \in \overline{Z}$. For example, if $X = Y = \mathbb{R}$ and $Z = \{x \mid x < 0\}$, then $\langle -x, Z, x \rangle$ is the absolute value function. In the intended application, the functions in question will be acts (i.e., functions from the set of states S to consequences C). So $a = \langle a_1, X, a_2 \rangle$ is the act such that $a(s) = a_1(s)$ for all $s \in X$ and $a(s) = a_2(s)$ for all $s \in \overline{X}$. For brevity, we identify the consequence $c \in C$ with the constant act a_c such that $a_c(s) = c$ for all $s \in S$. So for $c_1, c_2 \in C$, $\langle c_1, X, c_2 \rangle$ is the act with the property that $a(s) = c_1$ for all $s \in X$ and $a(s) = c_2$ for all $s \in \overline{X}$. Recall

that $X_1, \dots, X_n$ is a *partition* of Y iff the X_i 's are nonempty and pairwise disjoint, and $\bigcup_i X_i = Y$.

Fix some decision situation (A, S, C) . Readers familiar with [Savage, 1954] will recall that Savage implicitly assumes that A consists of all possible functions from S to C , since the DM can be questioned about any pair of functions. Throughout this paper, we have assumed that A could be any nonempty subset of the set of all simple acts. It is possible to maintain that assumption here, though some of the postulates would fail, not because $\lesssim_A$ does not relate certain members of A , but because A does not contain certain pairs of acts. We could adapt the postulates by “relativizing” them (so acts not in A are not required to be related) as we do in the full paper; however, that involves changing the statements of Savage’s postulates somewhat, and the presentation becomes more complicated. To simplify the exposition here, we assume in this section only that A consists of *all* simple acts. Here are Savage’s first six postulates (stated under the assumption that A consists of all simple acts from S to C , so that, for example, if a and b are in A , and $X \subseteq S$, then $\langle a, X, b \rangle \in A$):

- P1. For all $a_1, a_2, a_3 \in A$,
- (a) $a_1 \lesssim_A a_2$ or $a_2 \lesssim_A a_1$, and
 - (b) if $a_1 \lesssim_A a_2$ and $a_2 \lesssim_A a_3$, then $a_1 \lesssim_A a_3$.
- P2. For all $a_1, a_2, b_1, b_2 \in A$ and $X \subseteq S$,
- $\langle a_1, X, b_1 \rangle \lesssim_A \langle a_2, X, b_1 \rangle$ iff $\langle a_1, X, b_2 \rangle \lesssim_A \langle a_2, X, b_2 \rangle$.
- P3. For all $X \subseteq S$, if there exist some $a_1, a_2 \in A$ such that for all $b \in A$, $\langle a_1, X, b \rangle \prec_A \langle a_2, X, b \rangle$, then for all $c_1, c_2 \in C$, $c_1 \lesssim_A c_2$ iff
- for all $a \in A$, $\langle c_1, X, a \rangle \prec_A \langle c_2, X, a \rangle$, or
 - for all $a \in A$, $\langle c_1, X, a \rangle \sim_A \langle c_2, X, a \rangle$.
- P4. For all $X_1, X_2 \subseteq S$, $c_1, d_1, c_2, d_2 \in C$, if $d_1 \prec_A c_1$ and $d_2 \prec_A c_2$, then
- $\langle c_1, X_1, d_1 \rangle \lesssim_A \langle c_1, X_2, d_1 \rangle$ iff $\langle c_2, X_1, d_2 \rangle \lesssim_A \langle c_2, X_2, d_2 \rangle$.
- P5. There exist $c_1, c_2 \in C$ such that $c_1 \prec_A c_2$.
- P6. For all $a, b \in A$, $c \in C$, if $a \prec_A b$, then there exists a partition $Z_1, \dots, Z_n$ of S , such that for all Z_i , $\langle c, Z_i, a \rangle \prec_A b$ and $a \prec_A \langle c, Z_i, b \rangle$.

P1 is the standard necessary condition for representation by EU (and many of its generalizations), since the reals are linearly ordered; it basically says that $\lesssim_A$ is a total preorder. More specifically, P1(a) says that $\lesssim_A$ is total (from which reflexivity follows) and P1(b) says that it is transitive. P2 is the *sure-thing principle*, which says that the way two acts are related depends only on their differences (that is, the part on which they agree can be ignored, since that is the “sure thing”). Savage defines for each subset $X \subseteq S$ a conditional preference relation as follows: $a_1 \lesssim_A^X a_2$ iff

- for all $a \in A$, $\langle a_1, X, a \rangle \prec_A \langle a_2, X, a \rangle$, or
- for all $a \in A$, $\langle a_1, X, a \rangle \sim_A \langle a_2, X, a \rangle$.

Intuitively, $a_1 \lesssim_A^X a_2$ iff when X occurs, the DM would find a_2 at least as good as a_1 . In the presence of P1, P2 holds iff $\lesssim_A^X$ is a total preorder for all X . Using $\lesssim_A^X$, Savage defines what it means for X to be null: X is *null* iff $a_1 \lesssim_A^X a_2$ for all $a_1, a_2 \in A$. P3 says that if X is not null, then $\lesssim_A^X$ and $\lesssim_A$ agree on the consequences. Savage defines a relation $\lesssim_S$ on events as follows: $X \lesssim_S Y$ iff for all $c, d \in C$, if $d \prec_A c$ then $\langle c, X, d \rangle \lesssim_A \langle c, Y, d \rangle$. Intuitively, we expect the DM to prefer a binary act that is more likely (according to her beliefs) to yield the more desirable consequence. P4, in the presence of P1–P3, basically ensures that $\lesssim_S$ is a total preorder. P5 asserts that S is not null. P1–P5 by themselves do not allow the construction of a unique EU representation.⁶ For this we need P6, which roughly says that for all pairs of acts $a, b \in A$ and consequences $c \in C$, if $a \prec_A b$ then we can partition S into events such that the DM’s preference about a and b is unaffected if c were to happen in any element of the partition. Savage also has a seventh postulate, but it is relevant only for general (nonsimple) acts; since we consider only simple acts, we omit it here.

Since we focus on additive decision problems in this section, we can define a notion of conditional GEU, which greatly simplifies the presentation of the GEU axioms. Given a plausibilistic decision problem $\mathcal{D} = ((A, S, C), E, \mathbf{u}, \text{P1})$ and $\emptyset \neq Z \subseteq S$, define the GEU of act a restricted to Z as follows:

$$\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a \upharpoonright Z) = \bigoplus_{x \in \mathbf{u}_a(Z)} \text{Pl}(\mathbf{u}_a^{-1}(x) \cap Z) \otimes x,$$

Note that $\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a \upharpoonright S) = \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a)$.

Since we are restricting to additive decision problems, it is easy to check that, given a partition $X_1, \dots, X_n$ of $Y \subseteq S$, we have that

$$\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a \upharpoonright Y) = \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a \upharpoonright X_1) \oplus \dots \oplus \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a \upharpoonright X_n).$$

Also, it is easy to check that for all nonempty proper subsets X of S ,

$$\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_{\langle a_1, X, a_2 \rangle}) = \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_{a_1} \upharpoonright X) \oplus \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_{a_2} \upharpoonright \overline{X}).$$

Let $\mathcal{E}_{\mathcal{D}}(X) = \{\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a \upharpoonright X) \mid a \in A\}$. (We omit the subscript $\mathcal{D}$ if it is clear from context.) Intuitively, $\mathcal{E}_{\mathcal{D}}(X)$ consists of all the expected utility values of acts in A restricted to X . To simplify the statement of one of the axioms, let $\langle\langle u, X, v \rangle\rangle = \text{Pl}(X) \otimes u \oplus \text{Pl}(\overline{X}) \otimes v$ if $\emptyset \neq X \neq S$, and let $\langle\langle u, \emptyset, v \rangle\rangle = v$ and $\langle\langle u, S, v \rangle\rangle = u$.

A1. For all $x, y, z \in \mathcal{E}(S)$,

- (a) $x \lesssim_V y$ or $y \lesssim_V x$, and
- (b) if $x \lesssim_V y$ and $y \lesssim_V z$, then $x \lesssim_V z$.

A2. For all $\emptyset \neq X \neq S$, $x_1, x_2 \in \mathcal{E}(X)$, $y_1, y_2 \in \mathcal{E}(\overline{X})$,

- $x_1 \oplus y_1 \lesssim_V x_2 \oplus y_2$ iff $x_1 \oplus y_2 \lesssim_V x_2 \oplus y_1$.

A3. For all $\emptyset \neq X \neq S$, if there exist $x_1, x_2 \in \mathcal{E}(X)$ such that for all $y \in \mathcal{E}(\overline{X})$ $x_1 \oplus y \prec_V x_2 \oplus y$, then for all $u_1, u_2 \in \text{ran}(\mathbf{u})$, $u_1 \lesssim_V u_2$ iff

⁶They do, however, ensure that $\lesssim_S$ is a *qualitative probability* (which is a binary relation on 2^S with certain properties) [Savage, 1954].

- for all $y \in \mathcal{E}(\overline{X})$,
 $\text{Pl}(X) \otimes u_1 \oplus y \prec_V \text{Pl}(X) \otimes u_2 \oplus y$, or
 - for all $y \in \mathcal{E}(\overline{X})$,
 $\text{Pl}(X) \otimes u_1 \oplus y \sim_V \text{Pl}(X) \otimes u_2 \oplus y$.
- A4. For all $X_1, X_2 \subseteq S$, $u_1, v_1, u_2, v_2 \in \text{ran}(\mathbf{u})$,
if $v_1 \prec_V u_1$ and $v_2 \prec_V u_2$, then
- $\langle\langle u_1, X_1, v_1 \rangle\rangle \lesssim_V \langle\langle u_1, X_2, v_1 \rangle\rangle$ iff
 $\langle\langle u_2, X_1, v_2 \rangle\rangle \lesssim_V \langle\langle u_2, X_2, v_2 \rangle\rangle$
- A5. For some $u_1, u_2 \in \text{ran}(\mathbf{u})$, $u_1 \prec_V u_2$.
- A6. For all $x, y \in \mathcal{E}(S)$, $u \in \text{ran}(\mathbf{u})$, if $x \prec_V y$, then
for all $a, b \in A$, $c \in C$, such that $\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a) = x$,
 $\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_b) = y$, and $\mathbf{u}(c) = u$, there exists a partition
 $Z_1, \dots, Z_n$ of S , such that x can be expressed as
 $x_1 \oplus \dots \oplus x_n$ and y can be expressed as $y_1 \oplus \dots \oplus y_n$,
where $x_i = \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a \upharpoonright Z_i)$ and $y_i = \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_b \upharpoonright Z_i)$ for
 $1 \leq i \leq n$, and for all $1 \leq k \leq n$,
- $\text{Pl}(Z_k) \otimes u \oplus \bigoplus_{i \neq k} x_i \prec_V y$, and
 - $x \prec_V \text{Pl}(Z_k) \otimes u \oplus \bigoplus_{i \neq k} y_i$.

A1 says that the expected utility values are linearly pre-ordered; more specifically, A1(a) says that they are totally preordered and A1(b) says that the relation is transitive. Note that A1 does not say that the whole valuation domain is linearly preordered: that would be a sufficient but not a necessary condition for GEU($\mathcal{D}$) to satisfy P1. Since we want necessary and sufficient conditions for our representation results, some axioms apply only to expected utility values rather than to arbitrary elements of the valuation domain. As the following theorem shows, every subset of $\{A1(a), A1(b), \dots, A6\}$ represents the corresponding subset of $\{P1(a), P1(b), \dots, P6\}$.

Theorem 4.1: For all $I \subseteq \{1(a), 1(b), \dots, 6\}$, $\{A_i \mid i \in I\}$ represents $\{P_i \mid i \in I\}$.

Theorem 4.1 is a strong representation result. For example, if we are interested in capturing all of Savage's postulates but the requirement that $\lesssim_A$ is totally ordered, and instead are willing to allow it to be partially ordered (a situation explored by Lehmann [1996]), we simply need to drop axiom A1(b). Although we have focused here on Savage's postulates, it is straightforward to represent many of the other standard postulates considered in the decision theory literature in much the same way.

5 Conclusion

We have introduced GEU, a notion of generalized EU and shown that GEU can (a) represent all preference relations on acts and (b) be customized to capture any subset of Savage's postulates. In [Chu and Halpern, 2003], we show that GEU can be viewed as a universal decision rule in an even stronger sense: almost every decision rule has a GEU representation. Thus, the framework of expectation structures together with GEU provides a useful level of abstraction in which to study the general problem of decision making and rules for decision making.

References

- [Benvenuti and Mesiar, 2000] P. Benvenuti and R. Mesiar. Integrals with respect to a general fuzzy measure. In M. Grabisch, T. Murofushi, and M. Sugeno, editors, *Fuzzy Measures and Applications—Theory and Applications*, pages 205–232. Physica Verlag, Heidelberg, 2000.
- [Chu and Halpern, 2003] F. Chu and J. Y. Halpern. Great expectations. Part II: Generalized expected utility as a universal decision rule. In *IJCAI '03*, 2003.
- [Fishburn, 1987] P. C. Fishburn. Reconsiderations in the foundations of decision under uncertainty. *The Economic Journal*, 97(388):825–841, 1987.
- [Friedman and Halpern, 1995] N. Friedman and J. Y. Halpern. Plausibility measures: a user's guide. In *Proc. 11th Conf. Uncertainty in AI*, pages 175–184. 1995.
- [Giang and Shenoy, 2001] P. H. Giang and P. P. Shenoy. A comparison of axiomatic approaches to qualitative decision making using possibility theory. In *Proc. 17th Conf. Uncertainty in AI*, pages 162–170, 2001.
- [Gilboa and Schmeidler, 1989] I. Gilboa and D. Schmeidler. Maxmin expected utility with a non-unique prior. *Journal of Mathematical Economics*, 18:141–153, 1989.
- [Gul, 1991] F. Gul. A theory of disappointment aversion. *Econometrica*, 59(3):667–686, 1991.
- [Lehmann, 1996] D. Lehmann. Generalized qualitative probability; Savage revisited. In *Proc. 12th Conf. Uncertainty in AI*, pages 318–388, 1996.
- [Luce, 1990] R. D. Luce. Rational versus plausible accounting equivalences in preference judgments. *Psychological Science*, 1:225–234, 1990.
- [Luce, 2000] R. D. Luce. *Utility of Gains and Losses: Measurement-Theoretical and Experimental Approaches*. Lawrence Erlbaum, London, 2000.
- [Neumann and Morgenstern, 1947] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 2nd edition, 1947.
- [Quiggin, 1993] J. Quiggin. *Generalized Expected Utility Theory: The Rank-Dependent Expected Utility Model*. Kluwer, 1993.
- [Resnik, 1987] M. D. Resnik. *Choices: An Introduction to Decision Theory*. University of Minnesota Press, 1987.
- [Savage, 1954] L. J. Savage. *Foundations of Statistics*. Wiley, 1954.
- [Schmeidler, 1989] D. Schmeidler. Subjective probability and expected utility without additivity. *Econometrica*, 57:571–587, 1989.
- [Yaari, 1987] M. E. Yaari. The dual theory of choice under risk. *Econometrica*, 55(1):95–115, 1987.

Great Expectations.

Part II: Generalized Expected Utility as a Universal Decision Rule*

Francis C. Chu and Joseph Y. Halpern

Department of Computer Science

Cornell University

Ithaca, NY 14853, U.S.A.

Email: {fcc,halpern}@cs.cornell.edu

Abstract

Many different rules for decision making have been introduced in the literature. We show that a notion of generalized expected utility proposed in a companion paper [Chu and Halpern, 2003] is a universal decision rule, in the sense that it can represent essentially all other decision rules.

1 Introduction

A great deal of effort has been devoted to studying decision making. A standard formalization describes the choices a decision maker (DM) faces as acts, where an *act* is a function from states to consequences. Many decision rules (that is, rules for choosing among acts, based on the tastes and beliefs of the DM) have been proposed in the literature. Some are meant to describe how “rational” agents should make decisions, while others aim at modeling how real agents actually make decisions. Perhaps the best-known approach is that of *maximizing expected utility* (EU). Normative arguments due to Savage [1954] suggest that rational agents should behave as if their tastes are represented by a real-valued utility function on the consequences, their beliefs about the likelihood of events (i.e., sets of states) are represented by a probability measure, and they are maximizing the expected utility of acts with respect to this utility and probability.

Despite these normative arguments, it is well known that EU often does not describe how people actually behave when they make decisions [Resnik, 1987]; thus EU is of limited utility if we want to model (and perhaps predict) how people will behave. As a result, many alternatives to EU have been proposed in the literature (see, for example, [Gul, 1991; Gilboa and Schmeidler, 1989; Giang and Shenoy, 2001; Quiggin, 1993; Schmeidler, 1989; Yaari, 1987]). Some of these rules involve representations of beliefs by means other than a (single) probability measure; in some cases, beliefs and tastes are combined in ways other than the standard way which produces expected utility; yet other cases, such as Maximin and Minimax Regret [Resnik, 1987], do not require a representation of beliefs at all.

*Work supported in part by NSF under grants IIS-0090145 and CTC-0208535 and by the DoD Multidisciplinary University Research Initiative (MURI) program administered by ONR under grant N00014-01-1-0795.

In a companion paper [Chu and Halpern, 2003], we propose a general framework in which to study and compare decision rules. The idea is to define a generalized notion of expected utility (GEU), where a DM’s beliefs are represented by plausibility measures [Friedman and Halpern, 1995] and the DM’s tastes are represented by general (i.e., not necessarily real-valued) utility functions. We show there that every preference relation on acts has a GEU representation. Here we show that GEU is universal in a much stronger sense: we show that essentially all decision rules have GEU representations. The notion of representing one decision rule using another seems to be novel. Intuitively, decision rules are functions from tastes (and beliefs) to preference relations, so a representation of a decision rule is a representation of a *function*, not a preference relation.

Roughly speaking, given two decision rules $\mathcal{R}_1$ and $\mathcal{R}_2$, an $\mathcal{R}_1$ *representation* of $\mathcal{R}_2$ is a function τ that maps inputs of $\mathcal{R}_2$ to inputs of $\mathcal{R}_1$ that contain the same representation of tastes (and beliefs) such that $\mathcal{R}_1(\tau(x)) = \mathcal{R}_2(x)$. Thus, τ models, in a precise sense, a user of $\mathcal{R}_2$ as a user of $\mathcal{R}_1$, since τ preserves tastes (and beliefs). We show that a large collection of decision rules have GEU representations and characterize the collection. Essentially, a decision rule has a GEU representation iff it is *uniform* in a precise sense. It turns out that there are well-known decision rules, such as maximizing Choquet expected utility (CEU) [Schmeidler, 1989] (which essentially assumes that the DM is representing beliefs using a Dempster-Shafer belief function Bel , and then maximizing CEU with respect to Bel), that have no GEU representations. This is because τ is not allowed to modify the representation of the tastes (and beliefs). We then define a notion of *ordinal representation*, in which τ is allowed to modify the representation of the tastes (and beliefs), and is required to preserve only the ordinal aspect of the tastes (and beliefs). We show that almost all decision rules, including CEU, have ordinal GEU representations.

There seems to be no prior work in the literature that considers how one decision rule can represent another. Perhaps the closest results to our own are those of Lehmann [2001]. He proposes a “unified general theory of decision” that contains both quantitative and qualitative decision theories. He considers a particular decision rule he calls *Expected Qualitative Utility Maximization*, which allows utilities to be non-standard real numbers; he defines a certain preorder on the

nonstandard reals and makes decisions based on maximizing expected utility (with respect to that preorder). That his framework has EU as a special case is immediate, since for the standard reals, his preorder reduces to the standard order on the reals. He argues informally that Maximin is a special case of his approach, so that his approach can capture aspects of more qualitative decision making as well. It is easy to see that Lehmann's approach is a special case of GEU; his rule is clearly not universal in our sense.

2 Preliminaries

To make this paper self-contained, much of the material in the first three subsections of this section is taken (almost verbatim) from [Chu and Halpern, 2003].

2.1 Plausibility, Utility, and Expectation Domains

Since one of the goals of this paper is to provide a general framework for all of decision theory, we want to represent the tastes and beliefs of the DMs in as general a framework as possible. To this end, we use plausibility measures to represent the beliefs of the DMs and (generalized) utility functions to represent their tastes.

A *plausibility domain* is a set P , partially ordered by $\preceq_P$ (so $\preceq_P$ is a reflexive, antisymmetric, and transitive relation), with two special elements $\perp_P$ and $\top_P$, such that for all $x \in P$, $\perp_P \preceq_P x \preceq_P \top_P$. A function $\text{Pl} : 2^S \rightarrow P$ is a *plausibility measure* iff

$$\text{Pl1. } \text{Pl}(\emptyset) = \perp,$$

$$\text{Pl2. } \text{Pl}(S) = \top, \text{ and}$$

$$\text{Pl3. if } X \subseteq Y \text{ then } \text{Pl}(X) \preceq \text{Pl}(Y).$$

As pointed out in [Friedman and Halpern, 1995], plausibility measures generalize, not only probability, but a host of other representations of uncertainty as well. A *utility domain* is a set U endowed with a reflexive binary relation $\preceq_U$. Intuitively, elements of U represent the strength of likes and dislikes of the DM while elements of P represent the strength of her beliefs.

Once we have plausibility and utility, we want to combine them to form expected utility. To do this, we introduce expectation domains, which have utility domains, plausibility domains, and operators $\oplus$ (the analogue of $+$) and $\otimes$ (the analogue of $\times$).¹ More formally, an *expectation domain* is a tuple $E = (U, P, V, \otimes, \oplus)$, where $(U, \preceq_U)$ is a utility domain, $(P, \preceq_P)$ is a plausibility domain, $(V, \preceq_V)$ is a valuation domain (where $\preceq_V$ is a reflexive binary relation), $\otimes : P \times U \rightarrow V$, and $\oplus : V \times V \rightarrow V$. (As usual, we omit subscripts when they are clear.) We have four requirements on expectation domains:

$$E1. (x \oplus y) \oplus z = x \oplus (y \oplus z);$$

$$E2. x \oplus y = y \oplus x;$$

$$E3. \top \otimes x = x;$$

$$E4. (U, \preceq_U) \text{ is a substructure of } (V, \preceq_V).$$

¹Sometimes we use $\times$ to denote Cartesian product; the context will always make it clear whether this is the case.

$E1$ and $E2$ say that $\oplus$ is associative and commutative. $E3$ says that $\top$ is the left-identity of $\otimes$ and $E4$ ensures that the expectation domain respects the relation on utility values.

The *standard expectation domain*, which we denote $\mathbb{E}$, is $(\mathbb{R}, [0, 1], \mathbb{R}, +, \times)$, where the ordering on each domain is the standard order on the reals.

2.2 Decision Situations and Decision Problems

A *decision situation* describes the objective part of the circumstance that the DM faces (i.e., the part that is independent of the tastes and beliefs of the DM). Formally, a decision situation is a tuple $\mathcal{A} = (A, S, C)$, where

- S is the set of states of the world,
- C is the set of consequences, and
- A is a set of acts (i.e., a set of functions from S to C).

An act a is *simple* iff its range is finite. That is, a is simple if it has only finitely many consequences. Many works in the literature focus on simple acts (e.g., [Fishburn, 1987]). We assume in this paper that A contains only simple acts; this means that we can define (generalized) expectation using finite sums, so we do not have to introduce infinite series or integration for arbitrary expectation domains. Note that all acts are guaranteed to be simple if either S or C is finite, although we do not assume that here.

A decision problem is essentially a decision situation together with information about the tastes (and beliefs) of the DM; that is, a decision problem is a decision situation together with the subjective part of the circumstance that faces the DM. Formally, a *nonplausibilistic decision problem* is a tuple $(\mathcal{A}, U, \mathbf{u})$, where

- $\mathcal{A} = (A, S, C)$ is a decision situation,
- U is a utility domain, and
- $\mathbf{u} : C \rightarrow U$ is a utility function.

A *plausibilistic decision problem* is a tuple $(\mathcal{A}, E, \mathbf{u}, \text{Pl})$, where

- $\mathcal{A} = (A, S, C)$ is a decision situation,
- $E = (U, P, V, \otimes, \oplus)$ is an expectation domain,
- $\mathbf{u} : C \rightarrow U$ is a utility function, and
- $\text{Pl} : 2^S \rightarrow P$ is a plausibility measure.

We could have let a plausibilistic decision problem be simply a nonplausibilistic decision problem together with a plausibility domain and a plausibility measure, without including the other components of expectation domains. However, this turns out to complicate the presentation (see below).

We say that $\mathcal{D}$ is *standard* iff its utility domain is $\mathbb{R}$ (and, if $\mathcal{D}$ is plausibilistic, its plausibility measure is a probability measure and its expectation domain is $\mathbb{E}$).

2.3 Expected Utility

Let $\mathcal{D}$ be a decision problem with S as the set of states, U as the utility domain, and $\mathbf{u}$ as the utility function. Each act a of $\mathcal{D}$ induces a *utility random variable* $\mathbf{u}_a : S \rightarrow U$ as follows: $\mathbf{u}_a(s) = \mathbf{u}(a(s))$. If in addition $\mathcal{D}$ is plausibilistic with P as the plausibility domain and Pl as the plausibility measure,

then each a also induces a *utility lottery* $\ell_a^{\text{Pl}, \mathbf{u}} : \text{ran}(\mathbf{u}_a) \rightarrow P$ as follows: $\ell_a^{\text{Pl}, \mathbf{u}}(u) = \text{Pl}(\mathbf{u}_a^{-1}(u))$. If $\mathcal{D}$ is in fact standard (so $E = \mathbb{E}$ and Pl is a probability measure Pr), we can identify the expected utility of act a with the expected value of $\mathbf{u}_a$ with respect to Pr , computed in the standard way:

$$\mathbf{E}_{\text{Pr}}(\mathbf{u}_a) = \sum_{x \in \text{ran}(\mathbf{u}_a)} \text{Pr}(\mathbf{u}_a^{-1}(x)) \times x. \quad (2.1)$$

We can generalize (2.1) to an arbitrary expectation domain $E = (U, P, V, \otimes, \oplus)$ by replacing $+$, $\times$, and Pr by $\oplus$, $\otimes$, and Pl , respectively. This gives us

$$\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_a) = \bigoplus_{x \in \text{ran}(\mathbf{u}_a)} \text{Pl}(\mathbf{u}_a^{-1}(x)) \otimes x. \quad (2.2)$$

We call (2.2) the *generalized EU* (GEU) of act a . Clearly (2.1) is a special case of (2.2).

2.4 Decision Rules

Intuitively, a decision rule tells the DM what to do when facing a decision problem in order to get a preference relation on acts—e.g., compare the expected utility of acts. Just as we have nonplausibilistic decision problems and plausibilistic decision problems, we have nonplausibilistic decision rules and plausibilistic decision rules. As the name suggests, (non)plausibilistic decision rules are defined on (non)plausibilistic decision problems.

We do not require decision rules to be defined on all decision problems. For example, (standard) EU is defined only on standard plausibilistic decision problems. More formally, a (non)plausibilistic decision rule $\mathcal{R}$ is a function whose domain, denoted $\text{dom}(\mathcal{R})$, is a subcollection of the collection of (non)plausibilistic decision problems, and whose range, denoted $\text{ran}(\mathcal{R})$, is a collection of preference relations on acts. If $\mathcal{D} \in \text{dom}(\mathcal{R})$ and a_1 and a_2 are acts in $\mathcal{D}$, then we write

$$a_1 \lesssim_{\mathcal{R}(\mathcal{D})} a_2 \text{ iff } (a_1, a_2) \in \mathcal{R}(\mathcal{D}).$$

Here are a few examples of decision rules:

- GEU is a plausibilistic decision rule whose domain consists of all plausibilistic decision problems. Given a decision problem $\mathcal{D} = (\mathcal{A}, E, \dots)$, where $E = (U, P, V, \oplus, \otimes)$, we have $a_1 \lesssim_{\text{GEU}(\mathcal{D})} a_2$ iff $\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_{a_1}) \lesssim_V \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_{a_2})$ for all acts a_1, a_2 of $\mathcal{A}$. Note that GEU would not be a decision rule according to this definition if plausibilistic decision problems contained only a utility function and a plausibility measure, and did not include the other components of expectation domains.

- Of course, standard EU is a decision rule (whose domain consists of all standard plausibilistic decision problems).

- Maximin is a nonplausibilistic decision rule that orders acts according to their worst-case consequence. It is a conservative rule; the “best” act according to Maximin is the one with the best worst-case consequence. Intuitively, Maximin views Nature as an adversary that always pick a state that realizes the worst-case consequence, no matter what act the DM chooses. The domain of (standard) Maximin consists of nonplausibilistic decision problems with real-valued utilities. Given an act a and a real-valued utility function $\mathbf{u}$, let $\mathbf{w}_{\mathbf{u}}(a) = \min_{s \in S} \mathbf{u}_a(s)$. Then given a decision problem $\mathcal{D} = (\mathcal{A}, \mathbb{R}, \mathbf{u})$, $a_1 \lesssim_{\text{Maximin}(\mathcal{D})} a_2$ iff $\mathbf{w}_{\mathbf{u}}(a_1) \leq \mathbf{w}_{\mathbf{u}}(a_2)$.

- Minimax Regret (REG) is based on a different philosophy. It tries to hedge a DM’s bets, by doing reasonably well no matter what the actual state is. It is also a nonplausibilistic rule. As a first step to defining it, given a nonplausibilistic decision problem $\mathcal{D} = ((A, S, C), \mathbb{R}, \mathbf{u})$, for each state $s \in S$, let $\bar{u}_s = \sup_{a \in A} \mathbf{u}_a(s)$; that is, $\bar{u}_s$ is the least upper bound of the utilities in state s . The *regret* of a in state s , denoted $\mathbf{r}(a, s)$, is $\bar{u}_s - \mathbf{u}_a(s)$; note that no act can do better than a by more than $\mathbf{r}(a, s)$ in state s . Let $\bar{\mathbf{r}}(a) = \sup_{s \in S} \mathbf{r}(a, s)$. For example, suppose that $\bar{\mathbf{r}}(a) = 2$ and the DM picks a . Suppose that the DM then learns that the true state is s_0 and is offered a chance to change her mind. No matter what act she picks, the utility of the new act cannot be more than 2 higher than $\mathbf{u}_a(s_0)$. REG orders acts by their regret and thus takes the “best” act to be the one that minimizes $\bar{\mathbf{r}}(a)$. Intuitively, this rule tries to minimize the regret that a DM would feel if she discovered what the situation actually was: the “I wish I had done a_2 instead of a_1 ” feeling. Thus, $a_1 \lesssim_{\text{REG}(\mathcal{D})} a_2$ iff $\bar{\mathbf{r}}(a_1) \geq \bar{\mathbf{r}}(a_2)$. Like Maximin, Nature is viewed as an adversary that would pick a state that maximizes regret, no matter what act the DM chooses. It is well known that, in general, Maximin, REG, and EU give different recommendations [Resnik, 1987].

- The Maxmin Expected Utility rule (MMEU) [Gilboa and Schmeidler, 1989] assumes that a DM’s beliefs are represented by a set $\mathcal{P}$ of probability measures. Act a_1 is preferred to a_2 if the worst-case expected utility of a_1 (taken over all the probability measures in $\mathcal{P}$) is at least as large as the worst-case expected utility of a_2 . Thus MMEU is, in a sense, a hybrid of EU and Maximin. To view MMEU as a function on decision problems, we must first show how to represent a set of probability measures as a single plausibility measure. We do this using an approach due to Halpern [2001]. Let the plausibility domain $P = [0, 1]^{\mathcal{P}}$, that is, all functions from $\mathcal{P}$ to $[0, 1]$, ordered pointwise; in other words, $p \preceq_P q$ iff $p(\text{Pr}) \leq q(\text{Pr})$ for all $\text{Pr} \in \mathcal{P}$. Thus, in this domain, $\perp$ is the constant function 0 and $\top$ is the constant function 1. For each $X \subseteq S$, let $f_X \in P$ be the function that evaluates each probability measure in $\mathcal{P}$ at X ; that is, $f_X(\text{Pr}) = \text{Pr}(X)$ for all $\text{Pr} \in \mathcal{P}$. Let $\text{Pl}_{\mathcal{P}}(X) = f_X$; it is easy to verify that $\text{Pl}_{\mathcal{P}}$ is a plausibility measure. We view $\text{Pl}_{\mathcal{P}}$ as a representation of the set $\mathcal{P}$ of probability measures; clearly $\mathcal{P}$ can be recovered from $\text{Pl}_{\mathcal{P}}$. The domain of MMEU consists of all plausibilistic decision problems of the form $\mathcal{D} = ((A, S, C), (\mathbb{R}, [0, 1]^{\mathcal{P}}, V, \oplus, \otimes), \mathbf{u}, \text{Pl}_{\mathcal{P}})$, where $\mathcal{P}$ is a set of probability measures on 2^S , and $a_1 \lesssim_{\text{MMEU}(\mathcal{D})} a_2$ iff $\inf_{\text{Pr} \in \mathcal{P}} \mathbf{E}_{\text{Pr}}(\mathbf{u}_{a_1}) \leq \inf_{\text{Pr} \in \mathcal{P}} \mathbf{E}_{\text{Pr}}(\mathbf{u}_{a_2})$. Note that this definition ignores $\oplus$, $\otimes$, and V .

- (Dempster-Shafer) *belief functions* [Dempster, 1967] are a representation of uncertainty that generalize probability. That is, every probability measure is a belief function, but the converse is not necessarily true.² Given a belief function Bel , it is well-known that there exists a set $\mathcal{P}_{\text{Bel}}$ of probability measures such that for all $X \subseteq S$, $\text{Bel}(X) = \inf_{\text{Pr} \in \mathcal{P}_{\text{Bel}}} \text{Pr}(X)$ [Dempster, 1967]. A notion of expected

²Due to lack of space, we assume that the reader is familiar with belief functions.

utility for belief functions was defined by Choquet [1953] as follows. Given an act a and a real-valued utility function $\mathbf{u}$, if $\text{ran}(\mathbf{u}_a) = \{u_1, \dots, u_n\}$ and $u_1 < \dots < u_n$, then

$$\mathbf{E}_{\text{Bel}}(\mathbf{u}_a) = u_1 + \sum_{i=2}^n \text{Bel}(X_i) \times (u_i - u_{i-1}), \quad (2.3)$$

where $X_i = \mathbf{u}_a^{-1}(\{u_i, \dots, u_n\})$. It is easy to check (2.3) agrees with (2.1) if Bel is a probability measure. Moreover,

$$\mathbf{E}_{\text{Bel}}(\mathbf{u}_a) = \inf_{\text{Pr} \in \mathcal{P}_{\text{Bel}}} \mathbf{E}_{\text{Pr}}(\mathbf{u}_a) \quad (2.4)$$

in general. The Choquet expected utility (CEU) rule has as its domain decision problems of the form $\mathcal{D} = (\mathcal{A}, \mathbb{E}, \mathbf{u}, \text{Bel})$. Define $a_1 \preceq_{\text{CEU}(\mathcal{D})} a_2$ iff $\mathbf{E}_{\text{Bel}}(\mathbf{u}_{a_1}) \leq \mathbf{E}_{\text{Bel}}(\mathbf{u}_{a_2})$. It is immediate from (2.4) that if $\mathcal{D}_{\text{Bel}}$ is the decision problem that results from $\mathcal{D}$ by replacing Bel by Pl_{Bel} , and replacing the plausibility domain $[0, 1]$ in the expectation domain by $[0, 1]^{\mathcal{P}_{\text{Bel}}}$, then $a_1 \preceq_{\text{CEU}(\mathcal{D})} a_2$ iff $a_1 \preceq_{\text{MMEU}(\mathcal{D}_{\text{Bel}})} a_2$.

3 Representing Decision Rules

Given a decision rule $\mathcal{R}$ and a preference relation $\preceq_A$ on the set of acts A , an $\mathcal{R}$ representation of $\preceq_A$ is basically a decision problem $\mathcal{D} \in \text{dom}(\mathcal{R})$ such that $\mathcal{R}(\mathcal{D}) = \preceq_A$ (and the set of acts in $\mathcal{D}$ is A). In other words, an $\mathcal{R}$ representation of $\preceq_A$ makes $\mathcal{R}$ relate acts in A the way $\preceq_A$ relates them, so we can model a DM whose preference relation is $\preceq_A$ as a user of $\mathcal{R}$. In [Chu and Halpern, 2003] we show that every preference relation on acts has a GEU representation. Here we want to extend the notion of representation to decision rules; intuitively, we want an $\mathcal{R}_1$ representation of $\mathcal{R}_2$ to allow us to model a user of $\mathcal{R}_2$ as a user of $\mathcal{R}_1$. To make this precise, we need a few definitions.

Two (plausibilistic) decision problems $\mathcal{D}_1$ and $\mathcal{D}_2$ are *congruent*, denoted $\mathcal{D}_1 \cong \mathcal{D}_2$, iff they involve the same decision situation, utility domain, and utility function (and, if both are plausibilistic, the same plausibility domain and plausibility measure as well). Note that if $\mathcal{D}_1 \cong \mathcal{D}_2$, then they agree on the tastes (and beliefs) of the DM, so if they are both nonplausibilistic, then $\mathcal{D}_1 = \mathcal{D}_2$, and if they are both plausibilistic, then they differ only in the V , $\preceq_V$, $\oplus$, and $\otimes$ components of their expectation domains.

A *decision rule transformation* τ is a function that maps inputs of one decision rule $\mathcal{R}_2$ to the inputs of another rule $\mathcal{R}_1$. A decision rule transformation τ is an $\mathcal{R}_1$ representation of $\mathcal{R}_2$ iff $\text{dom}(\tau) = \text{dom}(\mathcal{R}_2)$ and for all $\mathcal{D} \in \text{dom}(\mathcal{R}_2)$,

$$\tau(\mathcal{D}) \cong \mathcal{D} \text{ and } \mathcal{R}_1(\tau(\mathcal{D})) = \mathcal{R}_2(\mathcal{D}).$$

Thus a DM that uses $\mathcal{R}_2$ to relate acts based on her tastes (and beliefs) behaves as if she is using $\mathcal{R}_1$, since $\tau(\mathcal{D}) \cong \mathcal{D}$ and $\mathcal{R}_1(\tau(\mathcal{D})) = \mathcal{R}_2(\mathcal{D})$.

Note that $\tau(\mathcal{D}) = \mathcal{D}$ is a GEU representation of EU. We now consider some less trivial examples.

Example 3.1: To see that Maximin has a GEU representation, let $E_{\text{max}} = (\mathbb{R}, \{0, 1\}, \mathbb{R}, \min, \times)$, and let Pl_{max} be the plausibility measure such that $\text{Pl}_{\text{max}}(U)$ is 0 if $X = \emptyset$ and 1 otherwise. If $\mathcal{D} = (\mathcal{A}, \mathbb{R}, \mathbf{u})$, where $\mathcal{A} = (A, S, C)$, then it is easy to check that $\mathbf{E}_{\text{Pl}_{\text{max}}, E_{\text{max}}}(\mathbf{u}_a) = \mathbf{w}_{\mathbf{u}}(a)$. Take $\tau(\mathcal{D}) = (\mathcal{A}, E_{\text{max}}, \mathbf{u}, \text{Pl}_{\text{max}})$. Clearly $\tau(\mathcal{D}) \cong \mathcal{D}$: the utility function has not changed. Moreover, it is immediate that $\text{GEU}(\tau(\mathcal{D})) = \text{Maximin}(\mathcal{D})$. ■

Example 3.2: To see that Minimax Regret (REG) has a GEU representation, for ease of exposition, we take $\text{dom}(\text{REG})$ to consist of standard decision problems $\mathcal{D} = ((A, S, C), \mathbb{R}, \mathbf{u})$ such that $M_{\mathcal{D}} = \sup_{a \in A, s \in S} \mathbf{u}_a(s) < \infty$. (If $M_{\mathcal{D}} = \infty$, given the restriction to simple acts, it is easy to show that all acts have infinite regret.) Let $E_{\text{reg}} = (\mathbb{R}, [0, 1], \mathbb{R}, \min, \otimes)$, where $x \otimes y = y - \log(x)$ if $x > 0$, and $x \otimes y = 0$ if $x = 0$. Note that $\perp = 0$ and $\top = 1$. Clearly, $\min$ is associative and commutative, and $\top \otimes r = r - \log(1) = r$ for all $r \in \mathbb{R}$. Thus, E_{reg} is an expectation domain.

For $\emptyset \neq X \subseteq S$, define $M_X = \sup_{a \in A, s \in X} \mathbf{u}_a(s)$. Note that $M_S = M_{\mathcal{D}} < \infty$; also if $X \subseteq Y$, then $M_X \leq M_Y$. Let $\text{Pl}_{\mathcal{D}}(\emptyset) = 0$ and $\text{Pl}_{\mathcal{D}}(X) = e^{M_X - M_S}$. It is easy to verify that $\text{Pl}_{\mathcal{D}}$ is a plausibility measure.

It is easy to check that $\mathbf{E}_{\text{Pl}_{\mathcal{D}}, E_{\text{reg}}}(\mathbf{u}_a) = M_{\mathcal{D}} - \mathbf{r}_{\mathbf{u}}(a)$ for all acts $a \in A$. Let $\tau(\mathcal{D}) = (\mathcal{A}, E_{\text{reg}}, \mathbf{u}, \text{Pl}_{\mathcal{D}})$. Since the utility function has not changed, $\tau(\mathcal{D}) \cong \mathcal{D}$; furthermore, higher expected utility corresponds to lower regret, so $\text{GEU}(\tau(\mathcal{D})) = \text{REG}(\mathcal{D})$. ■

Example 3.3: To see that MMEU has a GEU representation, let $\mathcal{D} = (\mathcal{A}, (\mathbb{R}, [0, 1]^{\mathcal{P}}, \widehat{V}, \widehat{\oplus}, \widehat{\otimes}), \mathbf{u}, \text{Pl}_{\mathcal{P}}) \in \text{dom}(\text{MMEU})$. Let $E_{\mathcal{P}} = (\mathbb{R}, [0, 1]^{\mathcal{P}}, \mathbb{R}^{\mathcal{P}}, \oplus, \otimes)$, where $\oplus$ is pointwise function addition, $\otimes$ is scalar multiplication, and

$$f \preceq_{\mathbb{R}^{\mathcal{P}}} g \text{ iff } \inf_{\text{Pr} \in \mathcal{P}} f(\text{Pr}) \leq \inf_{\text{Pr} \in \mathcal{P}} g(\text{Pr}).$$

Note that we can identify $\mathbb{R}$ with the constant functions in $\mathbb{R}^{\mathcal{P}}$, so $\mathbb{R}$ can be viewed as a substructure of $\mathbb{R}^{\mathcal{P}}$. With these definitions, $E_{\mathcal{P}}$ is an expectation domain.

Let $\tau(\mathcal{D}) = (\mathcal{A}, E_{\mathcal{P}}, \mathbf{u}, \text{Pl}_{\mathcal{P}})$. It is immediate from the definition of $\preceq_{\mathbb{R}^{\mathcal{P}}}$ that

$$a \preceq_{\text{GEU}(\tau(\mathcal{D}))} b \text{ iff } \inf_{\text{Pr} \in \mathcal{P}} \mathbf{E}_{\text{Pr}}(\mathbf{u}_a) \leq \inf_{\text{Pr} \in \mathcal{P}} \mathbf{E}_{\text{Pr}}(\mathbf{u}_b).$$

Thus $\text{GEU}(\tau(\mathcal{D})) = \text{MMEU}(\mathcal{D})$; furthermore, it is clear that $\tau(\mathcal{D}) \cong \mathcal{D}$, since the utility function and plausibility measure have not changed. ■

Although it can represent many decision rules, GEU cannot represent CEU. We can in fact characterize the conditions under which a decision rule is representable by GEU.

There is a trivial condition that a decision rule must satisfy in order for it to have a GEU representation. Intuitively, a decision rule $\mathcal{R}$ respects utility if $\mathcal{R}$ relates acts of constant utility according to the relation between utility values. Formally, a decision rule $\mathcal{R}$ respects utility iff for all $\mathcal{D} \in \text{dom}(\mathcal{R})$ with A as the set of acts, S as the set of states, U as the utility domain, and $\mathbf{u}$ as the utility function, for all $a_1, a_2 \in A$, if $\mathbf{u}_{a_i}(s) = u_i$ for all states $s \in S$, then

$$a_1 \preceq_{\mathcal{R}(\mathcal{D})} a_2 \text{ iff } u_1 \preceq_U u_2. \quad (3.5)$$

We say that $\mathcal{R}$ weakly respects utility iff (3.5) holds for all constant acts (but not necessarily for all acts of constant utility). It is easy to see that GEU respects utility, since $\top \otimes u = u$ for all $u \in U$ and $(U, \preceq_U)$ is a substructure of $(V, \preceq_V)$. Thus if $\mathcal{R}$ does not respect utility, it has no GEU representation. While respecting utility is a necessary condition for a decision rule to have a GEU representation, it is not sufficient. It is also necessary for the decision rule to treat acts that behave in similar ways similarly.

Two acts a_1, a_2 in a decision problem $\mathcal{D}$ are *indistinguishable*, denoted $a_1 \sim_{\mathcal{D}} a_2$ iff either

- $\mathcal{D}$ is nonplausibilistic and $\mathbf{u}_{a_1} = \mathbf{u}_{a_2}$, or
- $\mathcal{D}$ is plausibilistic, and $\ell_{a_1}^{\text{Pl}, \mathbf{u}} = \ell_{a_2}^{\text{Pl}, \mathbf{u}}$,

where $\mathbf{u}$ is the utility function of $\mathcal{D}$ and Pl is the plausibility measure of $\mathcal{D}$. In the nonplausibilistic case, two acts are indistinguishable if they induce the same utility random variable; in the plausibilistic case, they are indistinguishable if they induce the same utility lottery.

A decision rule $\mathcal{R}$ is uniform if it respects indistinguishability. More formally, $\mathcal{R}$ is *uniform* iff for all $\mathcal{D} \in \text{dom}(\mathcal{R})$ and a_1, a_2, b_1, b_2 acts of $\mathcal{D}$ such that $a_i \sim_{\mathcal{D}} b_i$,

$$a_1 \preceq_{\mathcal{R}(\mathcal{D})} a_2 \text{ iff } b_1 \preceq_{\mathcal{R}(\mathcal{D})} b_2.$$

Intuitively, we can think of utility random variables and utility lotteries as descriptions of what an act a does in terms of the tastes (and beliefs) of the DM. If $\mathcal{R}$ is uniform, we can view $\mathcal{R}$ as relating the acts indirectly by relating their descriptions.

As the following theorem shows, all uniform decision rules have GEU representations.

Theorem 3.4: *For all decision rules $\mathcal{R}$, $\mathcal{R}$ has a GEU representation iff $\mathcal{R}$ is uniform and $\mathcal{R}$ respects utility.*

Proof: The “if” direction is somewhat similar in spirit to the proof of Theorem 3.6, given below; due to the lack of space, we omit this direction.

For the “only if” direction, suppose that τ is a GEU representation of $\mathcal{R}$ and let $\mathcal{D}_0 \in \text{dom}(\mathcal{R})$ be arbitrary. Suppose that a_1, a_2, b_1, b_2 are acts of $\mathcal{D}_0$ such that $a_i \sim_{\mathcal{D}_0} b_i$. It is easy to check that if $\mathcal{D} = (\mathcal{A}, E, \mathbf{u}, \text{Pl}) \cong \mathcal{D}_0$, then $\mathbf{E}_{\text{Pl}, E}(\mathbf{u}_{a_i}) = \mathbf{E}_{\text{Pl}, E}(\mathbf{u}_{b_i})$. Thus for all plausibilistic $\mathcal{D}$, if $\mathcal{D} \cong \mathcal{D}_0$, then $a_1 \preceq_{\text{GEU}(\mathcal{D})} a_2$ iff $b_1 \preceq_{\text{GEU}(\mathcal{D})} b_2$. Since τ is a GEU representation of $\mathcal{R}$, $\tau(\mathcal{D}_0) \cong \mathcal{D}_0$ and $\mathcal{R}(\mathcal{D}_0) = \text{GEU}(\tau(\mathcal{D}_0))$. It follows then that $a_1 \preceq_{\mathcal{R}(\mathcal{D}_0)} a_2$ iff $b_1 \preceq_{\mathcal{R}(\mathcal{D}_0)} b_2$; thus $\mathcal{R}$ is uniform.

Now suppose that a_1 and a_2 are two acts of constant utility, say u_1 and u_2 , respectively, of $\mathcal{D}_0$. Since $\tau(\mathcal{D}_0) \cong \mathcal{D}_0$, a_i is still an act of constant utility u_i in $\tau(\mathcal{D}_0)$. Note that $a_1 \preceq_{\mathcal{R}(\mathcal{D}_0)} a_2$ iff $a_1 \preceq_{\text{GEU}(\tau(\mathcal{D}_0))} a_2$ iff $u_1 \preceq_U u_2$, where U is the utility domain of $\mathcal{D}_0$, since τ is a GEU representation of $\mathcal{R}$. Thus $\mathcal{R}$ respects utility. ■

Most of the decision rules we have discussed are uniform. However, CEU is not, as the following example shows:

Example 3.5: Let $\mathcal{D}_* = ((A, S, C), \mathbb{E}, \mathbf{u}, \text{Bel})$, where

- $A = \{a_1, a_2\}$; $S = \{s_1, s_2, s_3\}$; $C = \{1, 2, 3\}$;
- $\mathbf{u}(j) = j$, for $j = 1, 2, 3$;
- $a_1(s_j) = j$ and $a_2(s_j) = 3 - j$, for $j = 1, 2, 3$; and
- Bel is the belief function such that $\text{Bel}(X) = 1$ if $\{s_1, s_2\} \subseteq X$ and $\text{Bel}(X) = 0$ otherwise.

Since $\mathbf{u}_{a_i}^{-1}(j)$ is a singleton, $\text{Bel}(\mathbf{u}_{a_i}^{-1}(j)) = 0$ for $i = 1, 2$ and $j = 1, 2, 3$; thus $a_1 \sim_{\mathcal{D}_*} a_2$. On the other hand, by definition,

$$\mathbf{E}_{\text{Bel}}(\mathbf{u}_{a_1}) = 1 + \text{Bel}(X_{23})(2 - 1) + \text{Bel}(X_3)(3 - 2) = 1,$$

while

$$\mathbf{E}_{\text{Bel}}(\mathbf{u}_{a_2}) = 1 + \text{Bel}(X_{12})(2 - 1) + \text{Bel}(X_1)(3 - 2) = 2,$$

where $X_{ij} = \{s_i, s_j\}$ and $X_k = \{s_k\}$. It follows that CEU is not uniform, and so has no GEU representation. ■

The reader may have noticed an incongruity here. Example 3.3 shows that MMEU has a GEU representation; moreover, as shown earlier, MMEU produces essentially the same order on acts as CEU. However, CEU has no GEU representation. There is no contradiction to Theorem 3.4 here: There is no decision problem $\mathcal{D}$ such that $\mathcal{D} \cong \mathcal{D}_*$ (from Example 3.5) and $\text{GEU}(\mathcal{D}) = \text{CEU}(\mathcal{D}_*)$. However, $\text{GEU}((A, S, C), E_{\text{Pl}_{\text{Bel}}}, \mathbf{u}, \text{Pl}_{\text{Pl}_{\text{Bel}}}) = \text{CEU}(\mathcal{D}_*)$. Of course, $((A, S, C), E_{\text{Pl}_{\text{Bel}}}, \mathbf{u}, \text{Pl}_{\text{Pl}_{\text{Bel}}}) \not\cong \mathcal{D}_*$; $\text{Pl}_{\text{Pl}_{\text{Bel}}}$ and Bel are not the same, and they in fact represent related but different beliefs. (It is easy to show that sets are partially preordered by $\text{Pl}_{\text{Pl}_{\text{Bel}}}$ but totally preordered by Bel .)

The key reason that GEU cannot represent nonuniform decision rules is because they do not respect the indistinguishability relations imposed by the utility function (and the plausibility measure). Recall that we require that $\tau(\mathcal{D}) \cong \mathcal{D}$ because we want a user of one decision rules to appear as if she were using another, without pretending that she has different tastes (and beliefs). So we want τ to preserve the tastes (and beliefs) of its input.

There is a long-standing debate in the decision-theory literature as to whether preferences should be regarded as *ordinal* or *cardinal*. If they are ordinal, then all that matters is their order. If they are cardinal, then it should be meaningful to talk about the *differences* between preferences, that is, how much more a DM prefers one consequence to another. Similarly, if representations of likelihood are taken to be ordinal, then all that matters is whether one event is more likely than another. As we show below, if we require only that $\tau(\mathcal{D})$ and $\mathcal{D}$ describe the same ordinal tastes (and beliefs), then we can in fact express almost all decision rules, including CEU, in terms of GEU.

Two utility functions $\mathbf{u}_1 : C \rightarrow U_1$ and $\mathbf{u}_2 : C \rightarrow U_2$ represent the same ordinal tastes if for all $c_1, c_2 \in C$, $\mathbf{u}_1(c_1) \preceq_{U_1} \mathbf{u}_1(c_2)$ iff $\mathbf{u}_2(c_1) \preceq_{U_2} \mathbf{u}_2(c_2)$. Similarly, two plausibility measures $\text{Pl}_1 : 2^S \rightarrow P_1$ and $\text{Pl}_2 : 2^S \rightarrow P_2$ represent the same ordinal beliefs iff for all $X, Y \subseteq S$, $\text{Pl}_1(X) \preceq_{P_1} \text{Pl}_1(Y)$ iff $\text{Pl}_2(X) \preceq_{P_2} \text{Pl}_2(Y)$. Finally, two decision problems $\mathcal{D}_1$ and $\mathcal{D}_2$ are *similar*, denoted $\mathcal{D}_1 \simeq \mathcal{D}_2$, iff they involve the same decision situations, their utility functions represent the same ordinal tastes, and their plausibility measures represent the same ordinal beliefs. Note that $\mathcal{D}_1 \cong \mathcal{D}_2$ implies $\mathcal{D}_1 \simeq \mathcal{D}_2$, but the converse is false in general. A decision rule transformation τ is an *ordinal $\mathcal{R}_1$ representation of $\mathcal{R}_2$* iff $\text{dom}(\tau) = \text{dom}(\mathcal{R}_2)$ and for all $\mathcal{D} \in \text{dom}(\mathcal{R}_2)$, $\tau(\mathcal{D}) \simeq \mathcal{D}$ and $\mathcal{R}_1(\tau(\mathcal{D})) = \mathcal{R}_2(\mathcal{D})$.

We want to show next that almost all decision rules have an ordinal GEU representation. Doing so involves one more subtlety. Up to now, we have assumed that plausibility domains are partially ordered. This implies that two plausibility measures that represent the same ordinal beliefs necessarily induce the same indistinguishability relation (because of antisymmetry). Thus, in order to distinguish sets that have equivalent plausibilities when computing expected utility using $\otimes$ and $\otimes$, we need to allow plausibility domains to be partially *preordered*. So, for this result, we assume that $\preceq_P$ is a reflexive and transitive relation that is not necessarily antisymmetric (i.e., we could have that $p_1 \preceq_P p_2$ and $p_2 \preceq_P p_1$ but $p_1 \neq p_2$).

Theorem 3.6: A decision rule $\mathcal{R}$ has an ordinal GEU representation iff $\mathcal{R}$ weakly respects utility.

Proof: There are two cases, plausibilistic and nonplausibilistic. They are almost identical; we just do the plausibilistic case here. (Also, the “only if” direction is quite similar to the one in the proof of Theorem 3.4, so we omit it here.)

Suppose that $\mathcal{R}$ is a plausibilistic decision rule that weakly respects utility. Fix a plausibilistic decision problem $\mathcal{D} = ((A, S, C), E_1, \mathbf{u}_1, \text{Pl}_1) \in \text{dom}(\mathcal{R})$. Let U_1 and P_1 be the utility domain and plausibility domain of E_1 , respectively. Let $E_2 = (U_2, P_2, V, \oplus, \otimes)$ be defined as follows (note that $\times$ denotes Cartesian product in this proof):

- $U_2 = (U_1 \times C, \lesssim_{U_2})$, where
$$(u_1, c_1) \lesssim_{U_2} (u_2, c_2) \text{ iff } u_1 \lesssim_{U_1} u_2,$$
- $P_2 = (P_1 \times 2^S, \lesssim_{P_2})$, where
$$(p_1, X_1) \lesssim_{P_2} (p_2, X_2) \text{ iff } p_1 \lesssim_{P_1} p_2.$$

(Note that $\lesssim_{P_2}$ is a partial preorder, although it is not a partial order.)
- $V = (2^{S \times U_2}, \lesssim_V)$, where $x \lesssim_V y$ iff $x = y$ or
 1. $x = S \times \{(u_1, c_1)\}$, $y = S \times \{(u_2, c_2)\}$, and $(u_1, c_1) \lesssim_{U_2} (u_2, c_2)$, or
 2. $x = \{(s, (\mathbf{u}_1(a(s)), a(s))) \mid s \in S\}$,
 $y = \{(s, (\mathbf{u}_1(b(s)), b(s))) \mid s \in S\}$, and
 $a \lesssim_{\mathcal{R}(\mathcal{D})} b$, for some $a, b \in A$,
- $(p, X) \otimes (u, c) = X \times \{(u, c)\}$
- $x \oplus y = x \cup y$ for all $x, y \in V$.

Note that $(\perp_{P_1}, \emptyset) \lesssim_{P_2} (p, X) \lesssim_{P_2} (\top_{P_1}, S)$, so we have $\perp_{P_2} = (\perp_{P_1}, \emptyset)$ and $\top_{P_2} = (\top_{P_1}, S)$; thus, P_2 is a plausibility domain. Since $\mathcal{R}$ weakly respects utility, 1 and 2 are consistent with one another. We identify $(u, c) \in U_2$ with $S \times \{(u, c)\}$ in V ; with this identification, $\top \otimes (u, c) = (u, c)$ for all $(u, c) \in U_2$ and, given 1 in the definition of $\lesssim_V$, $(U_2, \lesssim_{U_2})$ is a substructure of $(V, \lesssim_V)$. Furthermore, $\oplus$ is clearly associative and commutative, so E_2 is indeed an expectation domain.

Now we need to define a utility function and a plausibility measure. Let $\mathbf{u}_2(c) = (\mathbf{u}_1(c), c)$ for all $c \in C$ and let $\text{Pl}_2(X) = (\text{Pl}_1(X), X)$ for all $X \subseteq S$. Note that

$$\text{Pl}_2(X) \lesssim_{P_2} \text{Pl}_2(Y) \text{ iff } \text{Pl}_1(X) \lesssim_{P_1} \text{Pl}_1(Y). \quad (3.6)$$

Thus Pl_2 is a plausibility measure, since Pl_1 is a plausibility measure. Also,

$$\mathbf{u}_2(c) \lesssim_{U_2} \mathbf{u}_2(d) \text{ iff } \mathbf{u}_1(c) \lesssim_{U_1} \mathbf{u}_1(d). \quad (3.7)$$

Let $\tau(\mathcal{D}) = ((A, S, C), E_2, \mathbf{u}_2, \text{Pl}_2)$. Note that, by (3.6) and (3.7), $\tau(\mathcal{D}) \simeq \mathcal{D}$; furthermore, it is easy to check that $\mathbf{E}_{\text{Pl}_2, E_2}((\mathbf{u}_2)_a) = \{(s, (\mathbf{u}_1(a(s)), a(s))) \mid s \in S\}$; it easily follows that $\text{GEU}(\tau(\mathcal{D})) = \mathcal{R}(\mathcal{D})$. Thus τ is an ordinal GEU representation of $\mathcal{R}$. ■

Theorem 3.6 shows that GEU can emulate essentially all decision rules. Thus, there is a sense in which GEU can be viewed as a universal decision rule. (We remark that although we have focused here on alternatives that are acts, in the sense of Savage, that is, functions from states to consequences, it is not hard to show—and we do in the full paper—that the same results hold if alternatives are taken to be horse lotteries, in the sense of Anscombe and Aumann [1963].)

References

- [Anscombe and Aumann, 1963] F. J. Anscombe and R. J. Aumann. A definition of subjective probability. *Annals of Mathematical Statistics*, 34(1):199–205, March 1963.
- [Choquet, 1953] G. Choquet. Theory of capacities. *Annales de l'Institut Fourier (Grenoble)*, 5:131–295, 1953.
- [Chu and Halpern, 2003] F. C. Chu and J. Y. Halpern. Great Expectations. Part I: On the customizability of generalized expected utility. In *Proc. Seventeenth International Joint Conference on Artificial Intelligence (IJCAI '01)*, pages 35–40, 2003.
- [Dempster, 1967] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38:325–339, 1967.
- [Fishburn, 1987] P. C. Fishburn. Reconsiderations in the foundations of decision under uncertainty. *The Economic Journal*, 97(388):825–841, December 1987.
- [Friedman and Halpern, 1995] N. Friedman and J. Y. Halpern. Plausibility measures: a user's manual. In *Proc. Eleventh Conference on Uncertainty in Artificial Intelligence (UAI '95)*, pages 175–184, 1995.
- [Giang and Shenoy, 2001] P. H. Giang and P. P. Shenoy. A comparison of axiomatic approaches to qualitative decision making using possibility theory. In *Proc. Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI 2001)*, pages 162–170, 2001.
- [Gilboa and Schmeidler, 1989] I. Gilboa and D. Schmeidler. Maxmin expected utility with a non-unique prior. *Journal of Mathematical Economics*, 18:141–153, 1989.
- [Gul, 1991] F. Gul. A theory of disappointment aversion. *Econometrica*, 59(3):667–686, May 1991.
- [Halpern, 2001] J. Y. Halpern. Conditional plausibility measures and Bayesian networks. *Journal of A.I. Research*, 14:359–389, 2001.
- [Lehmann, 2001] D. Lehmann. Expected qualitative utility maximization. *Games and Economic Behavior*, 35(1–2):54–79, April 2001.
- [Quiggin, 1993] J. Quiggin. *Generalized Expected Utility Theory: The Rank-Dependent Expected Utility Model*. Kluwer Academic Publishers, Boston, Mass., 1993.
- [Resnik, 1987] M. D. Resnik. *Choices: An Introduction to Decision Theory*. University of Minnesota Press, Minneapolis, 1987.
- [Savage, 1954] L. J. Savage. *Foundations of Statistics*. John Wiley & Sons, New York, 1954.
- [Schmeidler, 1989] D. Schmeidler. Subjective probability and expected utility without additivity. *Econometrica*, 57:571–587, 1989.
- [Yaari, 1987] M. E. Yaari. The dual theory of choice under risk. *Econometrica*, 55(1):95–115, January 1987.

Qualitative Decision under Uncertainty: Back to Expected Utility

Hélène Fargier

IRIT, 118 Route de Narbonne
31062 Toulouse Cedex, France
fargier@irit.fr

Régis Sabbadin

INRA-UBIA,
BP 27 - 31326 Castanet-Tolosan Cedex - France
sabbadin@toulouse.inra.fr

Abstract

Different qualitative models have been proposed for decision under uncertainty in Artificial Intelligence, but they generally fail to satisfy the principle of strict Pareto dominance or principle of "efficiency", in contrast to the classical numerical criterion — expected utility. In [Dubois and Prade, 1995] qualitative criteria based on possibility theory have been proposed, that are appealing but inefficient in the above sense. The question is whether it is possible to reconcile possibilistic criteria and efficiency. The present paper shows that the answer is yes, and that it leads to special kinds of expected utilities. It is also shown that although numerical, these expected utilities remain qualitative: they lead to two different decision procedures based on min, max and reverse operators only, generalizing the leximin and leximax orderings of vectors.

1 Introduction and motivation

A decision-making problem under uncertainty is a 4-tuple $(S, X, \mathcal{A}, \succeq)$, where S is a set of states of nature, X a set of consequences, $\mathcal{A} = X^S$ the set of possible acts (in decision under uncertainty, an act is a function $f : S \mapsto X$) and $\succeq$ is a *preference relation* on $\mathcal{A}$, usually complete and transitive (i.e. is a complete preorder).

A numerical approach is classically advocated (see e.g. [Savage, 1954]) for encoding both the information pertaining to the states of nature and the preferences on X : uncertainty is represented by a probability distribution p and preference is encoded by a utility function $u : X \rightarrow [0, 1]$ ¹. The pair $\langle p, u \rangle$ will be called a *probabilistic utility model*, PU-model for short. Acts are then ranked according to their expected utility $EU_{p,u}$ (written here in the finite setting):

$$f \succeq_{EU_{p,u}} g \Leftrightarrow EU_{p,u}(f) \geq EU_{p,u}(g)$$

where, $\forall h \in \mathcal{A} \quad EU_{p,u}(h) = \sum_{s \in S} p(s) \cdot u(h(s))$.

Information about preference and uncertainty in decision problems cannot always be quantified in a simple way, but only qualitative evaluations can sometimes be attained. As a

consequence, the topic of qualitative decision theory is a natural one to consider [Pearl, 1993; Dubois and Prade, 1995; Brafman and Tennenholtz, 1997; Dubois *et al.*, 1998b; Doyle and Thomason, 1999; Giang and Shenoy, 2000; Dubois *et al.*, 2000]. Giving up the quantification of utility and uncertainty has led to give up the expected utility (EU) criterion as well: the principle of most theories of qualitative decision making is to model uncertainty by an *ordinal* plausibility relation on events and preference by a *complete pre-ordering* on consequences. In [Dubois and Prade, 1995; Dubois *et al.*, 1998b] two qualitative criteria based on possibility theory, an optimistic and a pessimistic one, are proposed and axiomatized whose definitions only require a finite ordinal scale $L = \{0_L \leq \dots \leq 1_L\}$ for evaluating both utility and plausibility:

- $f \succeq_{OPT,\pi,\mu} g \Leftrightarrow U_{OPT,\pi,\mu}(f) \geq U_{OPT,\pi,\mu}(g)$ where $\forall h, U_{OPT,\pi,\mu}(h) = \max_{s \in S} \min(\pi(s), \mu(h(s)))$
- $f \succeq_{PES,\pi,\mu} g \Leftrightarrow U_{PES,\pi,\mu}(f) \geq U_{PES,\pi,\mu}(g)$ where $\forall h, U_{PES,\pi,\mu}(h) = \min_{s \in S} \max(n(\pi(s)), \mu(h(s)))$,

where $n : L \rightarrow L$ is the order reversing function of L , $\pi : S \rightarrow L$ is a normalized possibility distribution and $\mu : X \rightarrow L$ is a utility function on X . In the following, $\langle S, X, L, \pi, \mu \rangle$ will be called a qualitative possibilistic utility model (QPU-model) and we will assume S, X and L to be finite, as is generally the case in qualitative decision making.

The value $U_{PES,\pi,\mu}(f)$ is high only if f gives good consequences in every "rather plausible" state. This criterion generalizes the Wald criterion, which estimates the utility of an act by that of its worst possible consequence. $U_{PES,\pi,\mu}$ is thus "pessimistic" or "cautious", the pessimism being moderated by taking relative possibilities of states into account. On the other hand, $U_{OPT,\pi,\mu}$ is a mild version of the *maximax* criterion which is "optimistic", or "adventurous".

Although appealing from a qualitative point of view, possibilistic utilities suffer from a lack of decisiveness called the "drowning effect": when two acts give an identical and extreme (either good or bad) consequence in some plausible state, they may be undistinguished by these criteria, although they may give significantly different consequences in the other states. As a consequence the principle of Pareto dominance is not satisfied. That is it may be the case that $\forall s, \mu(f(s)) \geq \mu(g(s))$ and that $\exists s^*, \pi(s^*) > 0$ and $\mu(f(s^*)) > \mu(g(s^*))$ but $g \succeq f$.

¹Since expected utility is not sensitive to linear transformations of u , the choice of $[0, 1]$ as the range for u is made for convenience.

Example 1 Let $S = \{s_1, s_2\}$, $L = \{0, 1, 2, 3, 4, 5\}$. Let f and g be two acts whose utilities in states s_1 and s_2 are listed below, as well as the possibility degrees of the states. One can check that $U_{OPT, \pi, \mu}(f) = U_{OPT, \pi, \mu}(g) = 3$ and $U_{PES, \pi, \mu}(f) = U_{PES, \pi, \mu}(g) = 3$ although f strictly dominates g ($\mu(f(s_1)) = \mu(g(s_1))$ and f has a better consequence in s_2).

	s_1	s_2
f	3	4
g	3	1
π	5	2

Most of the qualitative approaches [Pearl, 1993; Dubois and Prade, 1995; Brafman and Tennenholtz, 1997; Giang and Shenoy, 2000], fail to satisfy Pareto dominance. But this is not the case within expected utility theory, since this model obeys the following *Sure-Thing Principle* (STP) that insures that identical consequences do not influence the relative preference between two acts:

$$\text{STP} : \forall f, g, h, h', fAh \succeq gAh \Leftrightarrow fAh' \succeq gAh',$$

where fAh denotes the act identical to f on $A \subseteq S$ and to h on $S \setminus A$. When $\succeq$ is complete and transitive, the principle of Pareto dominance is a direct consequence of the STP. So, is it possible to benefit from the STP in the possibilistic framework in order to satisfy the Pareto principle? Unfortunately, it can be shown that this is generally not possible:

Proposition 1 Let $\langle S, X, L, \pi, \mu \rangle$ be a QPU model.

$\succeq_{OPT, \pi, \mu}$ (or $\succeq_{PES, \pi, \mu}$) satisfies the STP

$$\Leftrightarrow \exists ! s^* : \pi(s^*) = 1_L \text{ and } \forall s \neq s^*, \pi(s) = 0_L.$$

This means that possibilistic decision criteria cannot obey the STP, except in a very particular case: when the actual state of the world is known, i.e. when there is no uncertainty at all! So, we cannot stay in the pure QPU framework and escape the drowning effect altogether. The idea is then to cope with the difficulty by proposing *refinements* of the possibilistic criteria that obey the Sure Thing Principle².

This paper shows (Section 2) that any possibilistic model can be refined by an expected utility. The kind of expected utility that is at work, and the very special probability measure that underlies it, are studied in Section 3 under the light of related work. It is also shown (Section 4) that although numerical, these expected utility criteria remain qualitative, since they lead to a decision procedure based solely on min, max and reverse operators — these new procedures generalize well known leximin and leximax decision procedures.

2 Expected utility refinements of qualitative possibilistic utilities

Recall that a refinement $\succeq'$ of a relation $\succeq$ is a relation perfectly compatible with $\succeq$ (it agrees with $\succeq$ when $\succeq$ provides a strict preference), but can break ties by setting $f \succ' g$ for some f, g that are indifferent w.r.t. $\succeq$. Formally:

²The idea of refining QPU first appeared in [Dubois et al., 2000]: the principle was to break ties through an extra criterion (e.g. refining the pessimistic QPU by the optimistic QPU or by another max-norm aggregation). The use of a max operator kept the approach in an ordinal framework, but forbade the full satisfaction of the STP

Definition 1 (Refinement)

$\succeq'$ refines $\succeq \Leftrightarrow \forall f, g \in \mathcal{A}, f \succ g \Rightarrow f \succ' g$.

Since we are looking for complete and transitive relations it is natural to think of refinements based on expected utility. Savage [1954] has indeed shown that, as soon as a complete preorder is desired that satisfies the STP and some very natural axioms, the EU criterion is almost unavoidable. So, the question is: *are there any expected utility criteria that refine the possibilistic criteria?*

Let $\langle S, X, L, \pi, \mu \rangle$ be a QPU model. When considering the optimistic (resp. pessimistic) criterion, we are looking for a probability distribution p and a utility function u such that $\succeq_{EU, p, u}$ refines $\succeq_{OPT, \pi, \mu}$ (resp. $\succeq_{PES, \pi, \mu}$). The idea is to build the EU criteria by means of a transformation $\chi : L \rightarrow [0, 1]$ that maps π to a probability distribution:

Definition 2 (Probabilistic transformation of a scale)

Let $\langle S, X, L, \pi, \mu \rangle$ be a QPU model. A probabilistic transformation of L w.r.t. π is a mapping $\chi : L \rightarrow [0, 1]$ such that $\chi(0_L) = 0$ and $p = \chi \circ \pi$ is a probability distribution.

Notice the presence of the condition $\chi(0_L) = 0$ that expresses the fact that the impossibility of an event (represented by a degree of 0_L in possibility theory) is expressed by a null probability. But the most plausible events (possibility degrees of 1_L), obviously do not receive a probability degree of 1, since they may be mutually exclusive. Notice also that we are looking for a unique function χ for transforming L — both p and u will be built upon this transformation. This is due to the fact that we assume that preference and uncertainty levels are commensurate and belong to the same scale : it is thus natural to transform the degrees regardless whether they model uncertainty or preference.

Moreover, π and μ originally represent all the information available to the user, both in terms of uncertainty of the actual state of the world and preference over possible consequences. So, no undesirable arbitrary information should be introduced in the refined decision model and p and u must be as close as possible to the original information: we are looking for “unbiased” transformations of L . Formally:

Definition 3 (Unbiased transformation of a scale)

Let $\langle S, X, L, \pi, \mu \rangle$ be a QPU model and χ a probabilistic transformation of L . χ is unbiased iff $\forall \alpha, \alpha' \in L, \alpha \leq \alpha' \Leftrightarrow \chi(\alpha) \leq \chi(\alpha')$

As a consequence, using an unbiased χ ensures that π and $p = \chi \circ \pi$ (resp. μ and $u = \chi \circ \mu$) are ordinarily equivalent:

- $\forall s, s' \in S, p(s) \geq p(s') \Leftrightarrow \pi(s) \geq \pi(s')$,
- $\forall x, y \in X, u(x) \geq u(y) \Leftrightarrow \mu(x) \geq \mu(y)$.

2.1 Expected utility refinements of optimistic QPU

Let us first provide a tractable sufficient condition for a probabilistic transformation to generate an expected utility that refines $\succeq_{OPT, \pi, \mu}$:

Proposition 2 Let $\langle S, X, L, \pi, \mu \rangle$ be a QPU model and χ be a probabilistic transformation of L w.r.t. π . Also let **H** be the condition :

$$\mathbf{H} : \forall \alpha, \alpha', \beta \in L \text{ s.t. } \beta \geq \alpha > \alpha', \\ \chi(\alpha) \cdot \chi(\beta) > \chi(\alpha') \cdot \chi(\beta) + (|S| - 1) \cdot \chi(1_L) \cdot \chi(\alpha').$$

Then $\succeq_{EU, \chi \circ \pi, \chi \circ \mu}$ refines $\succeq_{OPT, \pi, \mu}$ whenever χ satisfies **H**.

H is a sufficient condition to generate an EU-refinement of the optimistic QPU (it is also a necessary one when every degree in L is attained by both π and μ). Importantly, there always exists a probabilistic transformation of L satisfying **H**. Let $n_i = |\{s \in S, \pi(s) = \alpha_i\}|$ and $N = |S| + 1$.

Proposition 3

The function $\chi : L = \{\alpha_0 < \dots < \alpha_k\} \rightarrow [0, 1]$ such that $\chi(\alpha_0) = 0$ and $\chi(\alpha_i) = \frac{v}{N^{2k-i}}, i = 1, k$ satisfies **H** $\forall v > 0$. $\chi \circ \pi$ is a probability distribution iff $v = (\sum_{i=1,k} \frac{n_i}{N^{2k-i}})^{-1}$.

In the sequel, χ^* will denote the function $\chi^*(\alpha_i) = \frac{v}{N^{2k-i}}$ obtained with $v = (\sum_{i=1,k} \frac{n_i}{N^{2k-i}})^{-1}$.

Example 2 Let us take the QPU model of Example 1, where $N = 2$, $L = \{0, 1, 2, 3, 4, 5\}$. $\chi^*(L)$ is the series $(0, \frac{v}{N^{16}}, \frac{v}{N^8}, \frac{v}{N^4}, \frac{v}{N^2}, \frac{v}{N})$ where $v = (\frac{1}{N} + \frac{1}{N^8})^{-1}$. So:

$$EU(f) = \chi^*(5) \cdot \chi^*(3) + \chi^*(4) \cdot \chi^*(2) = \frac{v^2}{N^5} + \frac{v^2}{N^{10}},$$

$$EU(g) = \chi^*(5) \cdot \chi^*(3) + \chi^*(2) \cdot \chi^*(1) = \frac{v^2}{N^5} + \frac{v^2}{N^{24}},$$

f is thus preferred to g .

χ^* is in fact sufficient to generate every unbiased EU-refinement of $\succeq_{OPT, \pi, \mu}$, since all such refinements are equivalent.

Definition 4 Two relations $\succeq$ and $\succeq'$ are said to be equivalent ($\succeq \equiv \succeq'$) iff $\forall f, g \in \mathcal{A}, f \succeq g \Leftrightarrow f \succeq' g$.

Proposition 4 Let $\langle S, X, L, \pi, \mu \rangle$ be a QPU model, χ_1, χ_2 two unbiased probabilistic transformations of L w.r.t. π . $\succeq_{EU, \chi_1 \circ \pi, \chi_1 \circ \mu}$ and $\succeq_{EU, \chi_2 \circ \pi, \chi_2 \circ \mu}$ both refine $\succeq_{OPT, \pi, \mu}$ $\Rightarrow \succeq_{EU, \chi_1 \circ \pi, \chi_1 \circ \mu} \equiv \succeq_{EU, \chi_2 \circ \pi, \chi_2 \circ \mu}$

Notice that Proposition 4 does not mean that the numbers attached to the states by $p_1 = \chi_1 \circ \pi$ and $p_2 = \chi_2 \circ \pi$, nor the ones attached to the consequences by $u_1 = \chi_1 \circ \mu$ and $u_2 = \chi_2 \circ \mu$ are the same - it only means that the two models are ordinally equivalent, that they make the same decisions and order the events and the consequences in the same way. It also implies that the refinements that does not belong to this class (they may exist, e.g. those which introduce a total order in S or in X) cannot be unbiased : they must either introduce a strict preference between equivalent consequences or order equi-plausible states.

So, we get the following result for optimistic QPU models:

Theorem 1 For any QPU model $\langle S, X, L, \pi, \mu \rangle$:

- There exists an unbiased probabilistic transformation χ^* of L w.r.t. π such that $\succeq_{EU, \chi^* \circ \pi, \chi^* \circ \mu}$ refines $\succeq_{OPT, \pi, \mu}$.
- If χ and χ' are two unbiased transformations of L s.t. both $\succeq_{EU, \chi \circ \pi, \chi \circ \mu}$ and $\succeq_{EU, \chi' \circ \pi, \chi' \circ \mu}$ refine $\succeq_{OPT, \pi, \mu}$, then $\succeq_{EU, \chi \circ \pi, \chi \circ \mu} \equiv \succeq_{EU, \chi' \circ \pi, \chi' \circ \mu}$.

We have hence obtained what we were looking for: for any QPU model we are able to propose an EU model that refines $\succeq_{OPT, \pi, \mu}$. As a refinement, it is perfectly compatible with but more decisive than the optimistic utility. Moreover, it does not use other information than the original one - it is unbiased. Since based on expected utility, it obviously satisfies the Sure Thing Principle as well as Pareto Dominance.

2.2 EU refinements of pessimistic QPU

When considering the pessimistic qualitative model, the same kind of result can be obtained, noticing that $\succeq_{PES, \pi, \mu}$ and $\succeq_{OPT, \pi, \mu}$ are dual relations:

Proposition 5 Let $\langle S, X, L, \pi, \mu \rangle$ be a QPU model. Then: $\forall f, g \in \mathcal{A}, f \succeq_{PES, \pi, \mu} g \Leftrightarrow g \succeq_{OPT, \pi, \mu} f$.

Proposition 6 Let $\langle S, X, L, \pi, \mu \rangle$ be a QPU model and χ be a probabilistic transformation of L w.r.t. π . Let $p = \chi \circ \pi$, $u = \chi \circ \mu$, $u' = \chi(1_L) - \chi \circ n \circ \mu$. It holds that: $\succeq_{EU, p, u}$ refines $\succeq_{OPT, \pi, \mu}$ iff $\succeq_{EU, p, u'}$ refines $\succeq_{PES, \pi, \mu}$.

Consequently, it is always possible to build a probabilistic transformation χ^* using Theorem 1, a probability $p = \chi^* \circ \pi$ and a utility function $u' = \chi^*(1_L) - \chi^* \circ n \circ \mu$ that define an unbiased EU-refinement of $\succeq_{PES, \pi, \mu}$. This provides the following pessimistic counterpart of Theorem 1:

Theorem 2 For any QPU model $\langle S, X, L, \pi, \mu \rangle$:

- There exist at least one unbiased transformation χ^* of L w.r.t. π , a probability distribution $p = \chi^* \circ \pi$ and a utility function $u' = \chi^*(1_L) - \chi^* \circ n \circ \mu$ such that $\succeq_{EU, p, u'}$ refines $\succeq_{PES, \pi, \mu}$.
- Let χ and χ' be two unbiased probabilistic transformations of L . If both $\succeq_{EU, \chi \circ \pi, \chi(1_L) - \chi \circ n \circ \mu}$ and $\succeq_{EU, \chi' \circ \pi, \chi'(1_L) - \chi' \circ n \circ \mu}$ refine $\succeq_{PES, \pi, \mu}$, then they are equivalent.

At this point in the paper we have proved an important result for bridging qualitative possibilistic decision theory and expected utility theory: we have shown than any optimistic or pessimistic QPU model can be refined by a EU model. Thus, we may conclude that (i) possibilistic decision criteria are compatible with the classical expected utility criterion and (ii) choosing a EU model is advantageous, since it leads to a EU-refinement of the original rule (thus overcomes the lack of decisiveness of the possibilistic criteria), it satisfies the STP and the principle of Pareto. But this does not mean that qualitiveness and ordinality are given up. In Section 4, we will show that, although probabilistic and based on additive manipulations of numbers, these criteria remain ordinal. This is very natural: since we start with an ordinal model and do not accept any bias, we produce another (probabilistic but) ordinal model, in which the numbers only encode orders of magnitude — this is the topic of the next Section.

3 EU refinements and big-stepped probabilities : related work

Both EU refinements of Section 2 are based on the same transformation of the possibility π into a probability distribution $p = \chi \circ \pi$ ³. The corresponding measure P is actually a "big-stepped probability" that is, it satisfies:

³But the optimistic utility $u = \chi \circ \mu$ is not equal to the pessimistic utility $u' = \chi(1_L) - \chi \circ n \circ \mu$: u puts the emphasis on the best consequences, while the pessimistic u' provides a high utility when low consequences are avoided. It is actually hopeless to look for a common refinement of $\succeq_{OPT, \pi, \mu}$ and $\succeq_{PES, \pi, \mu}$ since it may happen that $f \succ_{OPT, \pi, \mu} g$ and $g \succ_{PES, \pi, \mu} f$ altogether.

Definition 5 A probability measure P is said to be big stepped iff: $\forall s \in S, P(\{s\}) > P(\{s' \text{ s.t. } P(\{s'\}) < P(\{s\})\})$

In other terms, for any $s, p(s) > \sum_{s' \text{ s.t. } p(s') < p(s)} p(s')$. Such measures are often encountered in the AI literature. First, they have much in common with [Spohn, 1990]’s κ – functions: these disbelief degrees can indeed be interpreted as the order of magnitude of a ϵ probability [Pearl, 1993; Giang and Shenoy, 1999], which is obviously a big stepped probability. Moreover, big stepped probabilities also form a special class of *lexicographic probabilities* in the sense of [Blume et al., 1991; Lehmann, 1998] — we add the restriction that here all the states within a single cluster are equiprobable. Indeed each cluster corresponds to a class of equipossible states and since we are looking for unbiased transformations, equipossibility leads to equiprobability. Finally, Definition 5 generalizes the notion of big-stepped probability of [Snow, 1999; Benferhat et al., 1999] — which is recovered when each cluster is a singleton. Big stepped probabilities have also been proposed by [Dubois et al., 1998a] as a way to refine any possibility/necessity measure ⁴.

This reasoning on the order of magnitude also applies to utility: in a discrete setting, big-stepped utilities can be defined in the same way:

Definition 6 A utility measure u is said to be big stepped iff: $\forall x, x' \in X, u(x) > u(x') \implies u(x) > (|S| - 1) \cdot u(x')$

The utility functions $\chi^* \circ \mu$ and $\chi^*(1_L) - \chi^* \circ n \circ \mu$ are big stepped utilities. It is also the case of the ϵ utilities that underly κ -utility functions [Pearl, 1993; Wilson, 1995; Bonet and Geffner, 1996; Giang and Shenoy, 2000]. These works have advocated an approach to decision under uncertainty based on κ -functions, but without taking the STP into account (decision is made on the order of magnitude only, with a criterion comparable to optimistic utility). The present work makes a step further: in order to satisfy Pareto optimality, we go back to the underlying ϵ utilities and probabilities, using *double* exponents for epsilons instead of simple ones — we remain “big stepped” on the join scale. The other contribution of our approach is that it can be followed to encode pessimistic utilities as well.

[La Valle and Fishburn, 1992; Hammond, 1998; Lehmann, 1998] have studied decision models of *lexicographic probabilities* or *lexicographic utilities*, but in these models, the lexicographic characteristic is used only on one of the two dimensions (either the likelihood level, or the utility level). We operate on both dimensions simultaneously using a join transformation.

4 QPU refinements are qualitative

Although probabilistic and based on additive manipulations of utilities, our EU criteria remain ordinal, as paradoxical as it may seem at first sight. To establish this claim, this Section relates the previous EU criteria to the ordinal comparison of

⁴The “probabilistic likelihood relation” $p = \chi \circ \pi$ is a refinement of both the necessity (N) and possibility (Π) relations based on π . Reciprocally, it is easy to show that, for any big stepped probability P , there exists a π such that P refines both Π and N .

vectors. When S is finite, the comparison of acts can indeed be seen as a comparison of vectors of pairs of elements of L :

Definition 7 The representative vector of any act $f \in \mathcal{A}$ is the vector $\vec{f} = ((\pi_1, \mu_1), \dots, (\pi_i, \mu_i), \dots, (\pi_N, \mu_N))$ where π_i stands for $\pi(s_i)$ and μ_i for $\mu(f(s_i))$.

Comparing acts thus amounts to comparing elements of $(L^2)^N$. For instance, $\succeq_{OPT, \pi, \mu}$ is a restriction to the case $M = 2$ of the general $\succeq_{Maxmin}$ relation on $(L^M)^N$.

Definition 8 (Maxmin relation) Let $\vec{u}, \vec{v} \in (L^M)^N$. Then $\vec{u} \succeq_{Maxmin} \vec{v} \iff \max_{i=1, N} \min_{j=1, M} u_{i,j} \geq \max_{i=1, N} \min_{j=1, M} v_{i,j}$

where $w_{i,j}$ is the j^{th} element of the i^{th} vector of $\vec{w} \in (L^M)^N$

If $\vec{u}$ and $\vec{v}$ are representative of some acts, $M = 2$ and it is obvious that $f \succeq_{OPT, \pi, \mu} g \iff \vec{f} \succeq_{Maxmin} \vec{g}$. In this Section we will propose a refinement of $\succeq_{Maxmin}$, based only on the ordinal comparison of degrees and we will show the equivalence between this purely syntactical decision rule and the above EU models.

4.1 Case of total ignorance

Let us first consider the degenerate case of total ignorance, where $\forall s \in S, \pi(s) = 1_L$. In this case, the comparison of acts comes down to the comparison of utility degrees: $\vec{f} = ((1_L, \mu_1), \dots, (1_L, \mu_N))$ becomes $\vec{f} = (\mu_1, \dots, \mu_N)$. So, $f \succeq_{OPT, \pi, \mu} g \iff \vec{f} \succeq_{Max} \vec{g}$ and $f \succeq_{PES, \pi, \mu} g \iff \vec{f} \succeq_{Min} \vec{g}$. In decision making, the comparison of vectors by the max and min operators is well known, as it is known that it suffers from a lack of decisive power. That is why refinements of $\succeq_{min}$ and $\succeq_{max}$ have been proposed [Moulin, 1988]:

Definition 9 (Leximax, Leximin) Let $\vec{u}, \vec{v} \in L^N$. Then

- $\vec{u} \succeq_{lmax} \vec{v} \iff (\forall j, u_{(j)} = v_{(j)} \text{ or } \exists i, \forall j < i, u_{(j)} = v_{(j)} \text{ and } u_{(i)} > v_{(i)})$
- $\vec{u} \succeq_{lmin} \vec{v} \iff (\forall j, u_{(j)} = v_{(j)} \text{ or } \exists i, \forall j > i, u_{(j)} = v_{(j)} \text{ and } u_{(i)} > v_{(i)})$

where, for any $\vec{w} \in L^N$, $w_{(k)}$ is the k -th biggest element of $\vec{w}$ (i.e. $w_{(1)} \geq \dots \geq w_{(N)}$).

In practice, the leximin (resp. leximax) comparison consists in ordering both vectors in increasing (resp. decreasing) order and then lexicographically comparing them.

Example 3 Let $\vec{u} = (3, 2, 4)$ and $\vec{v} = (2, 2, 4)$: $\vec{u} \succ_{lmax} \vec{v}$ since $u_{(1)} = v_{(1)} = 4$ and $u_{(2)} = 3 > v_{(2)} = 2$; $\vec{u} \succ_{lmin} \vec{v}$ since $u_{(3)} = v_{(3)} = 2$ and $u_{(2)} = 3 > v_{(2)} = 2$.

It is obvious that $\succeq_{lmax}$ refines $\succeq_{max}$ and $\succeq_{lmin}$ refines $\succeq_{min}$. Moreover, both relations escape the drowning effect and are very efficient: the only pairs of ties are vectors that are identical up to a permutation of their elements.

4.2 General Case

Since the leximax and leximin comparisons are good candidates in a particular case, we have imagined an extension of these procedures to the case of 2 dimensions $((L^M)^N$ instead of L^N). The only thing that we need is to use any complete preorder $\succeq$ on vectors of L^M instead of the classical relation $\geq$ on L . It is then possible to order the sub-vectors of any $\vec{w}$ according to $\succeq$ and to apply any of the previous procedures:

Definition 10 (Leximax($\succeq$), Leximin($\succeq$)) Let $\succeq$ be a complete preorder on L^M , $\cong$ the associated equivalence relation and $\triangleright$ is the strict part of $\succeq$. Let $\vec{u}, \vec{v} \in (L^M)^N$. Then

- $\vec{u} \succeq_{\text{leximax}(\succeq)} \vec{v} \Leftrightarrow (\forall j, u_{(\succeq, j)} \cong v_{(\succeq, j)} \text{ or } \exists i \text{ s.t. } \forall j < i, u_{(\succeq, j)} \cong v_{(\succeq, j)} \text{ and } u_{(\succeq, i)} \triangleright v_{(\succeq, i)})$.
- $\vec{u} \succeq_{\text{leximin}(\succeq)} \vec{v} \Leftrightarrow (\forall j, u_{(\succeq, j)} \cong v_{(\succeq, j)} \text{ or } \exists i \text{ s.t. } \forall j > i, u_{(\succeq, j)} \cong v_{(\succeq, j)} \text{ and } u_{(\succeq, i)} \triangleright v_{(\succeq, i)})$.

where, for any $\vec{w} \in (L^M)^N$, $w_{(\succeq, i)}$ is the i^{th} biggest sub-vector of $\vec{w}$ according to $\succeq$.

The leximax procedure can in particular be applied to the preorder $\succeq = \succeq_{\text{leximin}}$. In practice, this comparison consists in first ordering the elements of each sub-vector in increasing order w.r.t $\succeq$, then in ordering the sub-vectors in decreasing order (w.r.t. $\succeq_{\text{leximin}}$). It is then enough to lexicographically compare the two new vectors of vectors.

Proposition 7 ($\succeq_{\text{leximax}(\succeq_{\text{leximin}})$ order)

- i) $\succeq_{\text{leximax}(\succeq_{\text{leximin}})$ is a complete preorder;
- ii) $\succeq_{\text{leximax}(\succeq_{\text{leximin}})$ refines $\succeq_{\text{maxmin}}$;
- iii) If $N = 1$, then $\succeq_{\text{leximax}(\succeq_{\text{leximin}}) \equiv \succeq_{\text{leximin}}$;
- iv) If $M = 1$, then $\succeq_{\text{leximax}(\succeq_{\text{leximin}}) \equiv \succeq_{\text{leximax}}$.

So, $\succeq_{\text{leximax}(\succeq_{\text{leximin}})$ is the refinement of $\succeq_{\text{maxmin}}$ we desire. Let us now compare representative vectors of acts using this relation (letting $M = 2$) – we get a refinement of $\succeq_{\text{OPT}, \pi, \mu}$.

Proposition 8 The relation $\succeq_{\text{leximax}(\succeq_{\text{leximin}})$ defined by $f \succeq_{\text{leximax}(\succeq_{\text{leximin}})} g \Leftrightarrow \vec{f} \succeq_{\text{leximax}(\succeq_{\text{leximin}})} \vec{g}$ refines $\succeq_{\text{OPT}, \pi, \mu}$.

Example 4 The representative vectors of f and g in Example 1 are: $\vec{f} = ((5, 3), (2, 4))$, $\vec{g} = ((5, 3), (2, 1))$. $(5, 3) \cong_{\text{leximin}} (5, 3)$, $(4, 2) \succ_{\text{leximin}} (2, 1)$, so $\vec{f} \succ_{\text{leximax}(\succeq_{\text{leximin}})} \vec{g}$.

The same kind of reasoning can be followed to refine $\succeq_{\text{PES}, \pi, \mu}$. The pessimistic utility of act f is $\min_{s \in S} \max(n(\pi(s)), \mu(f(s)))$. We need to refine a minimax procedure, and this can be done using $\succeq_{\text{leximin}(\succeq_{\text{leximax}})$. Since operator $\max$ does not apply to $(\pi(s), \mu(f(s)))$ but to $(n(\pi(s)), \mu(f(s)))$, we use the π -reverse vectors of acts:

Definition 11 The π -reverse vector of an act $f \in \mathcal{A}$ is $n(\vec{f}) = ((n(\pi(s_1)), \mu(f(s_1))), \dots, (n(\pi(s_N)), \mu(f(s_N))))$.

Proposition 9 The relation $\succeq_{\text{leximin}(\succeq_{\text{leximax}, n})}$ defined by $f \succeq_{\text{leximin}(\succeq_{\text{leximax}, n})} g \Leftrightarrow n(\vec{f}) \succeq_{\text{leximin}(\succeq_{\text{leximax}})} n(\vec{g})$

refines $\succeq_{\text{PES}, \pi, \mu}$.

So, the lexi-refinement of $\succeq_{\text{PES}, \pi, \mu}$ applies the leximin(leximax) comparison to the π -reverse vectors, while the refinement of $\succeq_{\text{OPT}, \pi, \mu}$ applies the leximax(leximin) comparison directly to the representative vectors. Both procedures are purely ordinal: the degrees in L are only compared using min, max and reverse operators — only their relative orders matter. Our final result is that these refinements are equivalent to the EU-refinements identified in Section 2.

Theorem 3 Let $\langle S, X, L, \pi, \mu \rangle$ be a QPU model and χ a probabilistic transformation of L w.r.t. π , $p = \chi \circ \pi$, $u = \chi \circ \mu$, $u' = \chi(1_L) - \chi \circ n \circ \mu$:

- i) $\succeq_{\text{EU}, p, u}$ refines $\succeq_{\text{OPT}, \pi, \mu} \Leftrightarrow \succeq_{\text{EU}, p, u} \equiv \succeq_{\text{leximax}(\succeq_{\text{leximin}})$.
- ii) $\succeq_{\text{EU}, p, u'}$ refines $\succeq_{\text{PES}, \pi, \mu} \Leftrightarrow \succeq_{\text{EU}, p, u'} \equiv \succeq_{\text{leximin}(\succeq_{\text{leximax}, n})}$.

So, the probabilistic refinements of possibilistic utilities are equivalent to purely comparative procedures: efficient QPU refinements are probabilistic but remain qualitative. Reciprocally, we can prove that the $\succeq_{\text{leximax}(\succeq_{\text{leximin}})$ and $\succeq_{\text{leximin}(\succeq_{\text{leximax}, n})}$ preference relations over vectors of vectors always admit a representation by a sum (on N) of products (on M), provided that L is discrete – but this is beyond the scope of this paper, that focuses on decision under uncertainty (where $M=2$).

5 Conclusion

The topic of Qualitative Decision Theory has received much attention in the past few years and several approaches, including QPU, have been proposed. This latter model forms a convenient framework for a qualitative expression of problems of decision under uncertainty. However, it suffers from a lack of decisiveness. We have proposed EU-based refinements of QPU which proved to be perfectly compatible with the original qualitative expression of knowledge and preferences: the only difference is that lexicographic (leximax(leximin) or leximin(leximax)) comparisons are used instead of maxmin or minmax. The axiomatization of these decision procedures is out of the scope of this paper. It actually consists in the 5 basic Savagean axioms, together with “mild” versions of the pessimism or optimism axioms of possibilistic utilities (see [Fargier and Sabbadin, 2003]).

κ -utility functions have also received much attention in AI but they too suffer from a lack of decisive power. Indeed, due to the $\min$ operator in $U_\kappa(f) = \min_{s \in S} \{\kappa(s) + \mu(f(s))\}$, they cannot satisfy Savage’s STP. We claim that $\succeq_{\kappa, \mu}$ can be EU-refined, in the same way as we did for QPU, and that the EU-refinement can be expressed syntactically by replacing the $\min$ operator by a leximin one. The proof of this conjecture, which would follow the same line as the one presented here, is left for future research.

The other extension of our work is to refine monotonic utilities [Dubois et al., 1998] a family of decision criteria that generalizes QPU and are based on the Sugeno integral. Monotonic Utilities admit a maxmin expression $\max_{s \in S} (\min(\sigma(F_s), \mu(f(s))))$, where $\sigma : 2^S \rightarrow L$ is a monotonic measure and $F_s = \{s' \in S \text{ s.t. } \mu(f(s')) \geq \mu(f(s))\}$. Again, we conjecture that replacing $\min$ and $\max$ with their lexicographical versions allows an efficient refinement and lead to Choquet-EU criteria [Gilboa, 1987].

Appendix

Most of the proofs are omitted for sake of brevity and can be found in [Fargier and Sabbadin, 2003] or at the following address : <http://www-bia.inra.fr/T/sabbadin/WEB/FargierSabbadin03Rap.html>.

We provide here the sketches of the most interesting ones.

Proof of Proposition 1 $\Leftarrow$ is trivial to show.

$\Rightarrow$ For any $x, y \in X$, $A \subseteq S$, let us denote xAy the act such that $xAy(s) = x$ if $s \in A$, $xAy(s) = y$ otherwise. Let x_0 and x_1 be two consequences such that $\mu(x_0) = 0_L$ and $\mu(x_1) = 1_L$.

Suppose there exist s_1, s_2 , $s_1 \neq s_2$, $\pi(s_1) \geq \pi(s_2) > 0_L$. Then, we have $x_1\{s_2\}x_0 \succ_{\text{OPT}, \pi, \mu} x_0s_1x_1$ and $x_1\{s_1, s_2\}x_0 \preceq_{\text{OPT}, \pi, \mu} x_1\{s_1\}x_0$, which violates the STP. So, if

$\succeq_{OPT, \pi, \mu}$ satisfies the STP then there exists at most one s^* such that $\pi(s^*) > 0_L$. Since π is normalized, s^* exists and $\pi(s^*) = 1_L$. In the pessimistic case, just notice that $x1 \succ_{PES, \pi, \mu} x0\{s_2\}x1$ whereas $x0\{s_1, s_2\}x1 \succeq_{PES, \pi, \mu} x0\{s_1\}x1$. $\square$

Sketch of the proof of Proposition 4. Let us denote $L_\mu = \{\alpha_i \in L, \exists x \in X, \mu(x) = \alpha_i\}$ — let us rank $L_\mu = (\alpha_0 < \dots < \alpha_i < \dots < \alpha_k)$ and $L_\pi = \{\beta_i \in L, \exists s \in S, \pi(s) = \beta_i\}$ — let us rank $L_\pi = (\beta_0 < \dots < \beta_i < \dots < \beta_k)$. It can be assumed (without loss of generality) that $L_\pi \subseteq L_\mu = L$.

Consider any $\alpha_i \in L$ and $\beta_j = \min\{\beta_i \in L_\pi, \beta_i \geq \alpha_i\}$. There exists a pair (s, x) where $\pi(s) = \beta_j, \mu(x) = \alpha_i$. Let us also denote $x0$ (resp. $x1$) a consequence of utility 0_L (resp. 1_L).

The proof of Proposition 4 is based on the observation that act $f : f(s) = x, f(s') = x0$ is always strictly preferred by $\succeq_{OPT, \pi, \mu}$ to act $g : g(s') = x1$ when $\pi(s') < \alpha_i$ and $g(s) = \alpha_i - 1$ when $\pi(s') \geq \alpha_i$.

From this, it can be shown that (i) the sub products, in the expression of the expected utility, are ranked in the same way by any χ and that (ii) the act with the biggest j -product is surely strictly preferred to the other act (whatever the values of the remaining terms).

Lemma 1 Let χ_1 and χ_2 two unbiased transformation of L refining $\succeq_{OPT, \pi, \mu}$. It holds that: $\forall \alpha, \alpha' \in L_\mu, \forall \beta, \beta' \in L_\pi$: $\chi_1(\alpha) \cdot \chi_1(\beta) > \chi_1(\alpha') \cdot \chi_1(\beta') \iff \chi_2(\alpha) \cdot \chi_2(\beta) > \chi_2(\alpha') \cdot \chi_2(\beta')$

Lemma 2 $f \succ_{EU, \chi \circ \pi, \chi \circ \mu} g \iff \exists j/\forall i < j, \chi(\pi_{\sigma f(i)}) \cdot \chi(\mu_{\sigma f(i)}) = \chi(\pi_{\sigma g(i)}) \cdot \chi(\mu_{\sigma g(i)})$ and $\chi(\pi_{\sigma f(j)}) \cdot \chi(\mu_{\sigma f(j)}) > \chi(\pi_{\sigma g(j)}) \cdot \chi(\mu_{\sigma g(j)})$.

Lemmas 1 and 2 together are enough to prove the equivalence of $\succeq_{EU, \chi_1 \circ \pi, \chi_1 \circ \mu}$ and $\succeq_{EU, \chi_2 \circ \pi, \chi_2 \circ \mu}$, whatever χ_1 and χ_2 : Indeed, $f \succ_{EU, \chi_1 \circ \pi, \chi_1 \circ \mu} g \iff$ (from Lemma 2) $\exists j/\forall i < j, \chi_1(\pi_{\sigma f(i)}) \cdot \chi_1(\mu_{\sigma f(i)}) = \chi_1(\pi_{\sigma g(i)}) \cdot \chi_1(\mu_{\sigma g(i)})$ and $\chi_1(\pi_{\sigma f(j)}) \cdot \chi_1(\mu_{\sigma f(j)}) > \chi_1(\pi_{\sigma g(j)}) \cdot \chi_1(\mu_{\sigma g(j)})$. But, from Lemma 1, the comparison of products are the same if we change χ_1 into χ_2 , so we get: $\exists j/\forall i < j, \chi_2(\pi_{\sigma f(i)}) \cdot \chi_2(\mu_{\sigma f(i)}) = \chi_2(\pi_{\sigma g(i)}) \cdot \chi_2(\mu_{\sigma g(i)})$ and $\chi_2(\pi_{\sigma f(j)}) \cdot \chi_2(\mu_{\sigma f(j)}) > \chi_2(\pi_{\sigma g(j)}) \cdot \chi_2(\mu_{\sigma g(j)})$, which implies (Lemma 2) that $f \succ_{EU, \chi_2 \circ \pi, \chi_2 \circ \mu} g$.

The same reasoning, simply swapping χ_1 and χ_2 implies $f \succ_{EU, \chi_2 \circ \pi, \chi_2 \circ \mu} g \implies f \succ_{EU, \chi_1 \circ \pi, \chi_1 \circ \mu} g$, from which we get the equivalence of $\succeq_{EU, \chi_1 \circ \pi, \chi_1 \circ \mu}$ and $\succeq_{EU, \chi_2 \circ \pi, \chi_2 \circ \mu}$. $\square$

References

- [Benferhat et al., 1999] S. Benferhat, D. Dubois, and H. Prade. Possibilistic and standard probabilistic semantics of conditional knowledge. *Journal of Logic and Computation*, 9:873–895, 1999.
- [Blume et al., 1991] L. Blume, A. Brandenburger, and E. Dekel. Lexicographic probabilities and choice under uncertainty. *Econometrica*, 59(1):61–79, 1991.
- [Bonet and Geffner, 1996] B. Bonet and H. Geffner. Arguing for decisions: A qualitative model of decision making. In *Proc. of UAI'96*, pages 98–105, 1996.
- [Brafman and Tennenholtz, 1997] R. Brafman and M. Tennenholtz. Modeling agents as qualitative decision makers. *Artificial Intelligence*, 94:217–268, 1997.
- [Doyle and Thomason, 1999] J. Doyle and R. H. Thomason. Background to qualitative decision theory. *AI Magazine*, 20(2):55–68, 1999.

- [Dubois and Prade, 1995] D. Dubois and H. Prade. Possibility theory as a basis for qualitative decision theory. In *Proc. of IJCAI'95*, pages 1925–1930, 1995.
- [Dubois et al., 1998a] D. Dubois, H. Fargier, and H. Prade. Possibilistic likelihood relations. In *Proc. of IPMU'98*, pages 1196–1203. EDK, 1998.
- [Dubois et al., 1998b] D. Dubois, L. Godo, H. Prade, and A. Zapico. Making decision in a qualitative setting: From decision under uncertainty to case-based reasoning. In *Proc. of KR'98*, pages 594–605, 1998.
- [Dubois et al., 1998c] D. Dubois, H. Prade, and R. Sabbadin. Qualitative decision theory with sugeno integrals. In *Proc. of UAI'98*, pages 121–128, 1998.
- [Dubois et al., 2000] D. Dubois, L. Godo, H. Prade, and A. Zapico. Advances in qualitative decision theory: Refined rankings. In *Proc. IBERAMIA'00*, pages 427–436, 2000.
- [Fargier and Sabbadin, 2003] H. Fargier and R. Sabbadin. Bridging the gap between qualitative decision theory and expected utility. Technical Report IRIT/03-08-R, IRIT, April 2003.
- [Giang and Shenoy, 1999] P. H. Giang and P. P. Shenoy. On transformations between probability and spohnian disbelief functions. In *Proc. of UAI'99*, pages 236–244, 1999.
- [Giang and Shenoy, 2000] P. H. Giang and P. P. Shenoy. A qualitative utility theory for spohn's theory of epistemic beliefs. In *Proc. of UAI'2000*, pages 220–227, 2000.
- [Gilboa, 1987] I. Gilboa. Expected utility with purely subjective non-additive probabilities. *Journal of Mathematical Economics*, 16:65–88, 1987.
- [Hammond, 1998] P. J. Hammond. *Consequentialism, non archimedean probabilities and lexicographic expected utility*, In: *The logic of strategy*. C. Bicchieri, R. Jeffrey and B. Skyrms, 1998.
- [La Valle and Fishburn, 1992] I. H. La Valle and P. C. Fishburn. State-independent subjective expected lexicographic utility. *Journal of Risk and Uncertainty*, 5:217–240, 1992.
- [Lehmann, 1998] D. Lehmann. Nonstandard numbers for qualitative decision making. In *Proc. of TARK'98*, pages 161–174, 1998.
- [Moulin, 1988] H. Moulin. *Axioms of Cooperative Decision Making*. Wiley, New-York, 1988.
- [Pearl, 1993] J. Pearl. From conditional oughts to qualitative decision theory. In *Proc. of UAI'93*, pages 12–20, 1993.
- [Savage, 1954] L.J. Savage. *The Foundations of Statistics*. Wiley, New York, 1954.
- [Snow, 1999] P. Snow. Diverse confidence levels in a probabilistic semantics for conditional logics. *Artificial Intelligence*, 113(1–2):269–279, 1999.
- [Spohn, 1990] W. Spohn. A general non-probabilistic theory of inductive reasoning. In *Proc. of UAI'90*, pages 315–322. Elsevier Science, 1990.
- [Wilson, 1995] N. Wilson. An order of magnitude calculus. In *Proc. of UAI'95*, pages 548–555, 1995.