ECAI 2006
G. Brewka et al. (Eds.)
IOS Press, 2006

An Efficient Upper Approximation for Conditional Preference

Nic Wilson[1]

Abstract. The fundamental operation of dominance testing, i.e., determining if one alternative is preferred to another, is in general very hard for methods of reasoning with qualitative conditional preferences such as CP-nets and conditional preference theories (CP-theories). It is therefore natural to consider approximations of preference, and upper approximations are of particular interest, since they can be used within a constraint optimisation algorithm to find some of the optimal solutions. Upper approximations for preference in CP-theories have previously been suggested, but they require consistency, as well as strong acyclicity conditions on the variables. We define an upper approximation of conditional preference for which dominance checking is efficient, and which can be applied very generally for CP-theories.

1 Introduction

A basic operation for a preference formalism is testing dominance, i.e., checking if one alternative is preferred to another. Unfortunately this has been shown to be very hard (PSPACE-complete) [7] in general for CP-nets [1, 2] and conditional preference theories (CP-theories) [13, 12], two formalisms for reasoning with qualitative and comparative conditional preferences; the cases where it is known to be feasible are of a very simple form [6, 2].

Dominance testing has particular importance for constrained optimisation; the algorithm for constrained optimisation given in [3] involves dominance testing if one requires more than one undominated solution of a set of constraints, and can involve a great many dominance checks; similar problems apply to the approach in [11]. This problem can be side-stepped by using dominance checking with respect to an upper approximation of preference (see e.g., [13]). If a solution is undominated with respect to the upper approximation it ensures that it will be undominated with respect to the preference relation; the algorithm of [3] amended in this way then generates *some* undominated solutions, but usually not all of them. Dominance testing with respect to such an upper approximation needs to be fast, since many such tests will typically be required. However, it is often not essential that the upper approximation be a close approximation, since we can usually afford to lose undominated solutions—in many situations there will be huge numbers of them, and so we would not be able to explicitly list them all even if we could find them.

Efficient upper approximations have been defined in [13, 12], but they require restrictive conditions on the CP-theory: consistency and strong acyclicity properties on the variables used in the conditional preference statements. However there are many natural situations when these conditions are not satisfied, even for simple examples, see e.g., [5, 10, 11, 7]. For larger problems, it will be inconvenient and often impractical to have to restrict a user's preference statements so that the CP-theory is of such a special form, firstly, because this may very well mean they cannot express the preferences they wish to, and, secondly, because checking consistency of a CP-net or a CP-theory can be very hard [7].

The main contribution of this paper is to define an upper approximation of conditional preference, for which dominance testing is efficient, and which can be applied very widely for CP-theories; i.e., without the strong acyclicity properties on variables, and without assuming consistency. The approximation is also a closer one than previous upper approximations.

Section 2 describes conditional preference theories, as defined in [13, 12] which is an approach for reasoning with conditional preferences generalising CP-nets and TCP-nets [4]; a new semantics in terms of total pre-orders is also given. Section 3 describes *pre-ordered search trees* with their associated total pre-orders. Our new upper approximation is defined as the intersection of all those total pre-orders satisfying the CP-theory which arise from a pre-ordered search tree. Section 4 presents a number of technical results giving equivalent forms of these definitions; this leads to a simple and efficient algorithm for testing dominance with respect to the upper approximation. Section 5 discusses comparisons with other upper approximations and constrained optimisation.

2 Conditional Preference Theories

Let V be a set of n variables. For each $X \in V$ let $\underline{X}$ be the set of possible values of X; we assume $\underline{X}$ has at least two elements. For subset of variables $U \subseteq V$ let $\underline{U} = \prod_{X \in U} \underline{X}$ be the set of possible assignments to set of variables U. The assignment to the empty set of variables is written $\top$. An *outcome* is an element of $\underline{V}$, i.e., an assignment to all the variables. For partial tuples $a \in \underline{A}$ and $u \in \underline{U}$, we may write $a \models u$, or say a *extends* u, if $A \supseteq U$ and $a(U) = u$, i.e., a projected to U gives u. More generally, we say that a *is compatible with* u if there exists outcome $\alpha \in \underline{V}$ extending both a and u, i.e., such that $\alpha(A) = a$ and $\alpha(U) = u$.

The language $\mathcal{L}_V$ (abbreviated to $\mathcal{L}$) consists of statements of the form $u : x > x' \, [W]$ where u is an assignment to set of variables $U \subseteq V$ (i.e., $u \in \underline{U}$), x, x' are different values of variable X, and $\{X\}, U$ and W are pairwise disjoint. Let $T = V - (\{X\} \cup U \cup W)$. Such a conditional preference statement φ is intended to represent that given u and any assignment to T, x is preferred to x' irrespective of the values of W. This informal meaning is captured by the set φ^* of pairs of outcomes $\{(tuxw, tux'w') : t \in \underline{T}, w, w' \in \underline{W}\}$, (with $(tuxw, tux'w')$ meaning that $tuxw$ is preferred to $tux'w'$) since u is satisfied in both outcomes $tuxw$ and $tux'w'$, and variable X has the value x in the first, and x' in the second, and they differ at most on

[1] Cork Constraint Computation Centre, Department of Computer Science, University College Cork, Cork, Ireland, n.wilson@4c.ucc.ie

$\{X\} \cup W$. We also say that $tux'w'$ is a *worsening swap from* $tuxw$ (with respect to φ). So, pairs (α, β) in φ^* are intended to represent a preference for α over β, and statement φ is intended as a compact representation of the preference information φ^*.

Subsets Γ of the language $\mathcal{L}$ are called *conditional preference theories (CP-theories)* [13]. For CP-theory Γ, define Γ^* to be $\bigcup_{\varphi \in \Gamma} \varphi^*$, which represents a set of preferences. We suppose here that preferences should be transitive, so it is then natural to define the associated order $\succ_\Gamma$, induced on $\underline{V}$ by Γ, to be the transitive closure of Γ^*. So α is preferred to β, i.e., $\alpha \succ_\Gamma \beta$, if and only if there is a sequence of worsening swaps from α to β (each with respect to some element of Γ; see [13]). Relation $\succeq_\Gamma$ is defined by $\alpha \succeq_\Gamma \beta$ if and only if either $\alpha = \beta$ or $\alpha \succ_\Gamma \beta$. If φ is the statement $u : x > x'\ [W]$ and $u \in \underline{U}$ we may write $u_\varphi = u$, $U_\varphi = U$, $x_\varphi = x$, $x'_\varphi = x'$, $W_\varphi = W$. If W is empty we may omit $[W]$, writing just $u : x > x'$.

We say that outcome β *dominates* α (with respect to $\succ_\Gamma$) if $\beta \succ_\Gamma \alpha$. Outcome α is said to be *undominated* (w.r.t.$\succ_\Gamma$) if there exists no outcome that dominates it. (Although in this paper we focus on this strong sense of *undominated*, of interest also are outcomes α which are dominated only by outcomes which α dominates.) Finding an undominated outcome of a CP-theory amounts to finding a solution of a CSP [12] (so checking if there exists an undominated outcome is an NP-complete problem). Solution α to a set of constraints C (involving variables V) is said to be undominated if it is not dominated by any other solution of C. Finding undominated solutions seems to be a harder problem.

This is a relatively expressive language of conditional preferences: CP-nets [1, 2] can be represented by a set of statements of the form $u : x > x'\ [W]$ with $W = \emptyset$ [13], and TCP-nets [4] can be represented in terms of such statements with $W = \emptyset$ or $|W| = 1$ [12]. Lexicographic orders can also be represented [13]. Other related languages for conditional preferences include those in [8, 9].

Example. Let V be the set of variables $\{X, Y, Z\}$ with domains as follows: $\underline{X} = \{\mathtt{x}, \bar{\mathtt{x}}\}$, $\underline{Y} = \{\mathtt{y}, \bar{\mathtt{y}}\}$ and $\underline{Z} = \{\mathtt{z}, \bar{\mathtt{z}}\}$. Let $\varphi_1 = \top : \mathtt{x} > \bar{\mathtt{x}}$, let $\varphi_2 = \mathtt{x} : \mathtt{y} > \bar{\mathtt{y}}$, let $\varphi_3 = \mathtt{y} : \mathtt{z} > \bar{\mathtt{z}}$, and let $\varphi_4 = \bar{\mathtt{x}} : \bar{\mathtt{z}} > \mathtt{z}$. Let CP-theory Γ be $\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$. This gives rise to the following preferences, where e.g., $\mathtt{xyz} \succ^3 \mathtt{xy\bar{z}}$ means that $\mathtt{xy\bar{z}}$ is a worsening swap from $\mathtt{xyz}$ with respect to φ_3 (in other words, $(\mathtt{xyz}, \mathtt{xy\bar{z}}) \in \varphi_3^*$): $\mathtt{xyz} \succ^3 \mathtt{xy\bar{z}} \succ^2 \mathtt{x\bar{y}\bar{z}} \succ^1 \mathtt{\bar{x}\bar{y}\bar{z}} \succ^4 \mathtt{\bar{x}\bar{y}z}$. We also have $\mathtt{xyz} \succ^2 \mathtt{x\bar{y}z} \succ^1 \mathtt{\bar{x}\bar{y}z}$; and $\mathtt{xyz} \succ^1 \mathtt{\bar{x}yz}$, and $\mathtt{xy\bar{z}} \succ^1 \mathtt{\bar{x}y\bar{z}}$. In addition there is the cycle $\mathtt{\bar{x}y\bar{z}} \succ^4 \mathtt{\bar{x}yz} \succ^3 \mathtt{\bar{x}y\bar{z}}$. The relation $\succ_\Gamma$ is the transitive closure of these orderings. The cycle shows that Γ is inconsistent (see below). But, despite this "localised" inconsistency (which only involves two outcomes), useful things can be said. In particular, there is a single undominated outcome: $\mathtt{xyz}$. Furthermore, if we add the constraint $(Y = \bar{\mathtt{y}}) \vee (Z = \bar{\mathtt{z}})$ then there are two undominated solutions: $\mathtt{xy\bar{z}}$ and $\mathtt{x\bar{y}z}$. In this case the constraint also "removes the inconsistency" in the following sense: $\succ_\Gamma$ restricted to solutions is acyclic.

Some general properties of relations

A relation $\succcurlyeq$ on a set A is formally defined to be a set of pairs, i.e., a subset of $A \times A$. We will often write $a \succcurlyeq b$ to mean $(a, b) \in \succcurlyeq$. Relation $\succ$ on set A is irreflexive if and only if for all $a \in A$, it is not the case that $a \succ a$. A pre-order $\succcurlyeq$ on A is a reflexive and transitive relation, i.e., such that $a \succcurlyeq a$ for all $a \in A$, and such that $a \succcurlyeq b$ and $b \succcurlyeq c$ implies $a \succcurlyeq c$. Elements a and b are said to be $\succcurlyeq$-equivalent if $a \succcurlyeq b$ and $b \succcurlyeq a$. We may write $\succ$ for the *strict part of* $\succcurlyeq$, i.e., the relation given by $a \succ b$ if and only if $a \succcurlyeq b$ and $b \not\succcurlyeq a$. (Note, however, that $\succ_\Gamma$ is not necessarily the strict part of $\succeq_\Gamma$.) A relation $\succcurlyeq$ is complete if for all $a \neq b$, either $a \succcurlyeq b$ or $b \succcurlyeq a$. Relation $\succcurlyeq$ is anti-symmetric if $a \succcurlyeq b$ and $b \succcurlyeq a$ implies $a = b$ (i.e., iff $\succcurlyeq$-equivalence is no more than equality). A partial order is an anti-symmetric pre-order. A total pre-order is a complete pre-order. If $\succcurlyeq$ is a total pre-order then $a \succ b$ if and only if $b \not\succcurlyeq a$. A total order is a complete partial order. We say that a relation is acyclic if its transitive closure is anti-symmetric, i.e., there are no cycles $a \succcurlyeq a' \cdots \succcurlyeq a$ for different $a, a', \ldots$. A relation $\succcurlyeq'$ *extends* (or, alternatively, *contains*) relation $\succcurlyeq$ if $\succcurlyeq' \supseteq \succcurlyeq$, i.e., if $a \succcurlyeq' b$ holds whenever $a \succcurlyeq b$ holds.

Semantics

Sequences of worsening swaps can be considered as a proof theory for this system of conditional preference. In [13] a semantics is given in terms of total orders (based on the semantics for CP-nets [2]); in the case of an inconsistent CP-theory, the semantic entailment relation becomes degenerate. To deal with this problem, [5] defines an extended semantics for CP-nets in terms of total pre-orders. We show how this semantics can be generalised to CP-theories, extending also the semantics in [13].

Let $\succcurlyeq$ be a total pre-order on $\underline{V}$. We say that $\succcurlyeq$ *satisfies* conditional preference statement φ if $\alpha \succcurlyeq \beta$ holds whenever β is a worsening swap from α w.r.t. φ; this is if and only if $\succcurlyeq$ extends the relation φ^*. We say that $\succcurlyeq$ satisfies a CP-theory Γ if it satisfies each element φ of Γ, i.e., if $\succcurlyeq$ extends the relation Γ^*. Because $\succcurlyeq$ is transitive, this holds if and only if $\succcurlyeq$ extends $\succ_\Gamma$.

For different outcomes α and β we say that $\Gamma \models' (\alpha, \beta)$ if $\alpha \succcurlyeq \beta$ holds for all total pre-orders $\succcurlyeq$ satisfying Γ. We also say, as in [13], that $\Gamma \models (\alpha, \beta)$ if $\alpha \succcurlyeq \beta$ holds for all total orders $\succcurlyeq$ satisfying Γ. We say that Γ *is consistent* if there exists some *total order* satisfying Γ.

The following theorem[2] shows that swapping sequences are complete with respect to the semantics based on total pre-orders. Also, in the case of consistent CP-theories, the two semantic consequence relations are equivalent.

Theorem 1 *Let $\Gamma \subseteq \mathcal{L}_V$ be a CP-theory and let $\alpha, \beta \in \underline{V}$ be different outcomes. Then (i) the relation $\succeq_\Gamma$ is the intersection of all total pre-orders satisfying Γ; (ii) $\Gamma \models' (\alpha, \beta)$ if and only if $\alpha \succ_\Gamma \beta$; (iii) Γ is consistent if and only if $\succ_\Gamma$ is irreflexive if and only if $\succeq_\Gamma$ is a partial order; (iv) if Γ is consistent then $\Gamma \models (\alpha, \beta)$ if and only if $\Gamma \models' (\alpha, \beta)$.*

The semantics suggests a general approach to finding an upper approximation of $\succ_\Gamma$: we consider a subset $\mathcal{M}$ of the set of all total pre-orders (which might be thought of as a set of "preferred" models) and define that α is preferred to β (with respect to this approximation) if $\alpha \succcurlyeq \beta$ for all $\succcurlyeq$ in $\mathcal{M}$ which satisfy Γ. (It is an "upper approximation" in the sense that the approximation contains the relation $\succ_\Gamma$.) We use this kind of approach in the next section.

3 Pre-ordered Search Trees

A pre-ordered search tree is a rooted directed tree (which we imagine being drawn with the root at the top, and children below parents). Associated with each node r in the tree is a variable Y_r, which is instantiated with a different value in each of the node's children (if it

[2] The proof of this result, and proofs of other results in the paper are included in a longer version of the paper available at the 4C website: http://www.4c.ucc.ie/.

has any), and also a pre-ordering $\geqslant_r$ of the values of Y_r. A directed edge in the tree therefore corresponds to an instantiation of one of the variables to a particular value. Paths in the tree from the root down to a leaf node correspond to sequential instantiations of different variables. We also associate with each node r a set of variables A_r which is the set of all variables $Y_{r'}$ associated to nodes r' above r in the tree (i.e., on the path from the root to r), and an assignment a_r to A_r corresponding to the assignments made to these variables in the edges between the root and r. The root node r^* has $A_{r^*} = \emptyset$ and $a_{r^*} = \top$, the assignment to the empty set of variables. Hence r' is a child of r if and only if $A_{r'} = A_r \cup \{Y_r\}$ (where $A_r \not\ni Y_r$) and $a_{r'}$ extends a_r (with an assignment to Y_r).

More formally, define a node r to be a tuple $\langle A_r, a_r, Y_r, \geqslant_r\rangle$, where $A_r \subseteq V$ is a set of variables, $a_r \in \underline{A_r}$ is an assignment to those variables, $Y_r \in V - A_r$ is another variable, and $\geqslant_r$ is a total pre-order on the set $\underline{Y_r}$ of values of Y_r. We make two restrictions on the choice of this total pre-order: firstly, it is assumed not to be the trivial complete relation on $\underline{Y}$; so there exists some $y, y' \in \underline{Y}$ with $y \not\geqslant_r y'$. We also assume that total pre-order $\geqslant_r$ satisfies the following condition (which we require so that the associated ordering on outcomes is transitive): if there exists a child of node r associated with instantiation $Y_r = y$, then y is not $\geqslant_r$-equivalent to any other value of Y, so that $y \geqslant_r y' \geqslant_r y$ only if $y' = y$. In particular, $\geqslant_r$ totally orders the values (of Y_r) associated with the children of r.

For outcome α, define the *path to* α to be the path from the root which includes all nodes r such that α extends a_r. To generate this, for each node r, starting from the root, we choose the child associated with the instantiation $Y_r = \alpha(Y_r)$ (there is at most one such child); the path finishes when there exists no such child.

Node r is said to ***decide*** *outcomes* α *and* β if it is the deepest node (i.e., furthest from the root) which is both on the path to α and on the path to β. Hence α and β both extend the tuple a_r (but they may differ on variable Y_r). We compare α and β by using $\geqslant_r$, where r is the unique node which decides α and β.

Each pre-ordered search tree σ has an associated ordering relation $\succcurlyeq_\sigma$ on outcomes which is defined as follows. Let $\alpha, \beta \in \underline{V}$ be outcomes. We define $\alpha \succcurlyeq_\sigma \beta$ to hold if and only if $\alpha(Y_r) \geqslant_r \beta(Y_r)$, where r is the node which decides α and β. We therefore then have that α and β are $\succcurlyeq_\sigma$-equivalent if and only if $\alpha(Y_r)$ and $\beta(Y_r)$ are $\geqslant_r$-equivalent; also: $\alpha \succ_\sigma \beta$ holds if and only if $\alpha(Y_r) >_r \beta(Y_r)$. This ordering is similar to a lexicographic ordering in that two outcomes are compared on the first variable on which they differ.

The definition implies immediately that $\succcurlyeq_\sigma$ is reflexive and complete; it can be shown easily that it is also transitive. Hence it is a total pre-order. We say that pre-ordered search tree σ *satisfies* conditional preference theory Γ iff $\succcurlyeq_\sigma$ satisfies Γ.

Definition of Upper Approximation. For a given CP-theory Γ we define the relation $\trianglerighteq_\Gamma$ (abbreviated to $\trianglerighteq$) as follows: $\alpha \trianglerighteq \beta$ holds if and only if $\alpha \succcurlyeq_\sigma \beta$ holds for all σ satisfying Γ (i.e, all σ such that $\succcurlyeq_\sigma$ satisfies Γ). Relation $\trianglerighteq$ is then the intersection of all pre-ordered search tree orders which satisfy Γ. The intersection of a set of reflexive and transitive relations containing $\succ_\Gamma$ is clearly reflexive and transitive, and contains $\succ_\Gamma$:

Proposition 1 *For any CP-theory* Γ*, the relation* $\trianglerighteq_\Gamma$ *is a pre-order which is an upper approximation of* $\succ_\Gamma$*, i.e., if* $\alpha \succ_\Gamma \beta$ *then* $\alpha \trianglerighteq_\Gamma \beta$*.*

This implies, in particular, that if outcomes α and β are incomparable with respect to $\trianglerighteq_\Gamma$ then they are incomparable with respect to $\succ_\Gamma$.

Example continued. Consider the pre-ordered search tree σ_1 with just one node, the root node $r = \langle\emptyset, \top, Y, \mathrm{y} > \bar{\mathrm{y}}\rangle$. Let α and β be any outcomes with $\alpha(Y) = y$ and $\beta(Y) = \bar{\mathrm{y}}$. We then have $\alpha \succ_{\sigma_1} \beta$ since the node r decides α and β, and $\alpha(Y) >_r \beta(Y)$. In particular, this implies that σ_1 satisfies φ_2. If outcomes γ and δ agree on variable Y, then γ and δ are $\succcurlyeq_{\sigma_1}$-equivalent because $\gamma(Y)$ and $\delta(Y)$ are $\geqslant_r$-equivalent since $\gamma(Y) = \delta(Y)$. Hence σ_1 satisfies φ_1, φ_3 and φ_4 and so satisfies Γ. This implies that $\beta \not\trianglerighteq_\Gamma \alpha$, so, in particular, $\mathrm{x}\bar{\mathrm{y}}\mathrm{z} \not\trianglerighteq \mathrm{x}\mathrm{y}\bar{\mathrm{z}}$.

Now consider the pre-ordered search tree σ_2 which has root node $\langle\emptyset, \top, X, \mathrm{x} > \bar{\mathrm{x}}\rangle$ with a single child node $\langle\{X\}, \mathrm{x}, Z, \mathrm{z} > \bar{\mathrm{z}}\rangle$. This also satisfies Γ. For example, to check that σ_2 satisfies φ_4 we can reason as follows: let α and β be any outcomes such that $(\alpha, \beta) \in \varphi_4^*$, so that $\alpha(X) = \beta(X) = \bar{\mathrm{x}}$, $\alpha(Y) = \beta(Y)$, $\alpha(Z) = \bar{\mathrm{z}}$ and $\beta(Z) = \mathrm{z}$. Then the root node $r' = \langle\emptyset, \top, X, \mathrm{x} > \bar{\mathrm{x}}\rangle$ decides α and β because its single child is associated with $X = \mathrm{x}$, which is incompatible with α and β. Now, $Y_{r'} = X$ and α and β agree on X so $\alpha(X)$ and $\beta(X)$ are $\geqslant_{r'}$-equivalent; in particular, $\alpha(X) \geqslant_{r'} \beta(X)$. Hence $\alpha \succcurlyeq_{\sigma_2} \beta$. Pre-ordered search tree σ_2 strictly prefers $\mathrm{x}\bar{\mathrm{y}}\mathrm{z}$ to $\mathrm{x}\mathrm{y}\bar{\mathrm{z}}$, which shows that $\mathrm{x}\mathrm{y}\bar{\mathrm{z}} \not\trianglerighteq \mathrm{x}\bar{\mathrm{y}}\mathrm{z}$. We've shown that both $\succ_\Gamma$-undominated solutions are also $\trianglerighteq$-undominated. In fact, in this case, $\trianglerighteq_\Gamma$ is actually equal to $\succ_\Gamma$.

4 Computation of Upper Bound on Preference

Outcome α is $\trianglerighteq$-preferred to β if and only if α is preferred to β in all pre-ordered search trees satisfying Γ. At first sight this definition looks computationally very unpromising for two reasons: (i) direct testing of whether a pre-ordered search tree satisfies Γ is not feasible, as Γ^* will typically contain exponentially many pairs; (ii) there will very often be a huge number of pre-ordered search trees satisfying Γ.

In this section we find ways of getting round these two problems. We first (Section 4.1) find simpler equivalent conditions for a pre-ordered search tree σ to satisfy Γ; then, in 4.2, we use the results of 4.1 to recast the problem of testing dominance with respect to the upper approximation, allowing a simple and efficient algorithm.

4.1 Equivalent conditions for σ to satisfy Γ

Consider a pre-ordered search tree σ, and let α be any outcome. Associated with the path to α is the sequence of variables $Y_1, \ldots, Y_k$ which are instantiated along that path (i.e., associated with the nodes on the path), starting with the root node. If W is a set of variables and X is a variable not in W, we say that "*on the path to* α*,* X *appears before any of* W" if the following condition holds: $Y_j \in W$ implies that $Y_i = X$ for some $i < j$, i.e., if some element of W occurs on the path then X occurs earlier on the path.

Proposition 2 *The following pair of conditions are necessary and sufficient for a pre-ordered search tree* σ *to satisfy the CP-theory* Γ*:*

(1) For any $\varphi \in \Gamma$ *and outcome* α *extending* u_φ*: on the path to* α*,* X_φ *appears before* W_φ*;*

(2) for any node r *and any* $\varphi \in \Gamma$ *with* $X_\varphi = Y_r$ *we have* $x_\varphi \geqslant_r x'_\varphi$ *if* u_φ *is compatible with* a_r*.*

Relation $\sqsupset_a^X$. Because $\geqslant_r$ is transitive, condition (2) can be written equivalently as: for all nodes r in σ, $\geqslant_r \supseteq \sqsupset_{a_r}^{Y_r}$, where $\sqsupset_a^X$ is defined to be the transitive closure of the set of pairs (x, x') of values of X over all statements $u : x > x'\,[W]$ in Γ such that u is compatible with a. Note that relation $\sqsupset_a^X$ is monotonic decreasing with respect to a: if b extends a (to variables not including X) then $\sqsupset_a^X$ contains $\sqsupset_b^X$; this is because if u_φ is compatible with b then u_φ is compatible with a.

Let $A \subseteq V$ be a set of variables, let $a \in \underline{A}$ be an assignment to A and let $Y \in V - A$ be some variable not in A. We say that Y is *a-**choosable*** if $X_\varphi \in A$ for all $\varphi \in \Gamma$ satisfying (a) u_φ compatible with a and (b) $W_\varphi \ni Y$. Note that if $W_\varphi = \emptyset$ for all $\varphi \in \Gamma$, as in CP-nets, then, for all a, every variable is a-choosable. If we are attempting to construct a node r of a pre-ordered search tree satisfying Γ, where a_r is the associated assignment, then we need to pick a variable Y_r which is a_r-choosable, because of Proposition 2(1). This condition has the following monotonicity property: suppose $A \subseteq B \subseteq V$, and $Y \notin B$; suppose also that b is an assignment to B extending assignment a to variables A. Then Y is b-choosable if Y is a-choosable.

Define pre-ordered search tree node r to satisfy Γ if Y_r is a_r-choosable and $\geqslant_r$ satisfies condition (2) above, i.e., $\geqslant_r \supseteq \sqsupset_{a_r}^{Y_r}$. It can be seen that Y_r is a_r-choosable for each node r in pre-ordered search tree σ if and only if condition (1) of Proposition 2 is satisfied. This leads to the following result.

Proposition 3 *A pre-ordered search tree σ satisfies Γ if and only if each node of σ satisfies Γ.*

4.2 Computation of upper bound on preference relation

In this section we consider a fixed conditional preference theory Γ. Suppose we are given outcomes α and β, and we wish to determine if $\beta \trianglerighteq \alpha$ or not. By definition, $\beta \not\trianglerighteq \alpha$ if and only if there exists a pre-ordered search tree σ satisfying Γ with $\beta \not\succcurlyeq_\sigma \alpha$, i.e., with $\alpha \succ_\sigma \beta$. Therefore, an approach to showing $\beta \not\trianglerighteq \alpha$ is to construct a pre-ordered search tree σ with $\alpha \succ_\sigma \beta$. The key is to construct the path from the root which will form the intersection of the path to α and the path to β; for each node r on this path we need to choose a variable Y_r with certain properties. If α and β differ on Y_r it needs to be possible to choose the local relation $\geqslant_r$ so that $\alpha(Y_r) >_r \beta(Y_r)$. If α and β agree on Y_r then we need to ensure that the local relation is such that this node can have a child. With this in mind we make the following definitions.

Suppose α and β are both outcomes which extend partial tuple $a \in \underline{A}$. Define variable Y to be ***pickable** given a with respect to* (α, β) if $Y \notin A$ and (i) Y is a-choosable; (iia) if $\alpha(Y) = \beta(Y)$ then $\alpha(Y)$ is not $\sqsupset_a^Y$-equivalent to any other value in $\underline{Y}$; (iib) if $\alpha(Y) \neq \beta(Y)$ then $\beta(Y) \not\sqsupset_a^Y \alpha(Y)$. If Y is pickable given a with respect to (α, β), and $\alpha(Y) \neq \beta(Y)$ then we say that *Y is **decisive** given a (with respect to (α, β)).*

The following lemma describes a key monotonicity property. It follows immediately from the previously observed monotonicity properties of being a-choosable, and of $\sqsupset_a^Y$.

Lemma 1 *Let α and β be two outcomes which both extend tuples a and b, where $a \in \underline{A}$ and $b \in \underline{B}$ and $A \subseteq B \subseteq V$ (so b extends a). Let Y be a variable not in B. If Y is pickable given a with respect to (α, β) then Y is pickable given b with respect to (α, β).*

A ***decisive sequence** (w.r.t. (α, β))* is defined to be a sequence $Y_1, \ldots, Y_k$ of variables satisfying the following three conditions:

— for $j = 1, \ldots, k-1$, $\alpha(Y_j) = \beta(Y_j)$,
— $\alpha(Y_k) \neq \beta(Y_k)$
— for $j = 1, \ldots, k$, Y_j is pickable given a_j (with respect to (α, β)), where a_j is α restricted to $\{Y_1, \ldots, Y_{j-1}\}$; in particular, Y_k is decisive given a_k.

Proposition 4 *There exists a decisive sequence w.r.t. (α, β) if and only if there exists a pre-ordered search tree σ satisfying Γ with $\alpha \succ_\sigma \beta$.*

Since $\beta \trianglerighteq \alpha$ holds if and only if there exists no pre-ordered search tree σ satisfying Γ with $\alpha \succ_\sigma \beta$, Proposition 4 implies the following result.

Proposition 5 *For outcomes α and β, $\beta \trianglerighteq \alpha$ holds if and only if there exists no decisive sequence with respect to (α, β).*

Therefore to determine if $\beta \trianglerighteq \alpha$ or not, we just need to check if there exists a decisive sequence $Y_1, \ldots, Y_k$. The monotonicity lemma implies that we do not have to be careful about the variable ordering: a variable which is pickable at one point in the sequence is still pickable at a later point; this means that we can choose, for each j, Y_j to be any pickable variable, knowing that we will not have to backtrack, as any previously available choices remain available later.

The following algorithm takes as input outcomes α and β and determines if $\beta \trianglerighteq \alpha$ or not.

procedure Is $\beta \trianglerighteq \alpha$?

if $\alpha = \beta$ **then** return **true** and **stop**;
for $j := 1, \ldots, n$
    let a_j be α restricted to $\{Y_1, \ldots, Y_{j-1}\}$;
    if there exists a variable which is decisive given a_j w.r.t. (α, β)
        then return **false** and **stop**;
    if there exists a variable which is pickable given a_j w.r.t. (α, β)
        then let Y_j be any such variable;
        else return **true** and **stop**.

The correctness of the algorithm follows easily from Proposition 5 and the monotonicity lemma.

Theorem 2 *Let Γ be a CP-theory, and let α and β be outcomes. The above procedure is correct, i.e., it returns **true** if $\beta \trianglerighteq_\Gamma \alpha$, and it returns **false** if $\beta \not\trianglerighteq_\Gamma \alpha$.*

Example continued. Now let $\alpha = \mathrm{x\bar{y}z}$ and $\beta = \mathrm{x\bar{y}\bar{z}}$. Since for all $\varphi \in \Gamma$, $W_\varphi = \emptyset$, each variable is a-choosable for any a. The relation $\sqsupset_\top^Z$ contains pair $(\mathrm{z}, \bar{\mathrm{z}})$ because of $\varphi_3 = \mathrm{y} : \mathrm{z} > \bar{\mathrm{z}}$ (anything is compatible with $a = \top$). It also contains pair $(\bar{\mathrm{z}}, \mathrm{z})$ because of $\varphi_4 = \bar{\mathrm{x}} : \bar{\mathrm{z}} > \mathrm{z}$ and so $\beta(Z) \sqsupset_\top^Z \alpha(Z)$ which implies that Z is not pickable given $\top$ with respect to (α, β) since $\alpha(Z) \neq \beta(Z)$. On the other hand, X and Y are both pickable given $\top$ (but not decisive). Suppose we select $Y_1 = Y$, and so $a_2 = \bar{\mathrm{y}}$. Variable Z is still not pickable, but X is still pickable given a_2 (by Lemma 1) so we get $Y_2 = X$ and $a_3 = \mathrm{x\bar{y}}$. Relation $\sqsupset_{a_3}^Z$ is empty so Z is now pickable and hence decisive (giving decisive sequence Y, X, Z) proving that $\beta \not\trianglerighteq \alpha$. A shorter decisive sequence is X, Z which corresponds to pre-ordered search tree σ_2 which strictly prefers α to β.

Complexity of approximate dominance checking We assume that the size $|\underline{X}|$ of the domain of each variable X is bounded by a constant; we will consider the complexity in terms of n, the number of variables, and of $m = |\Gamma|$, where we assume that m is at least $O(n)$. This algorithm can be implemented to have complexity at worst $O(mn^2)$, or, more precisely, $O(mn(w+1))$ where w is the average of $|W_\varphi|$ over $\varphi \in \Gamma$. Clearly $w < n$. For some special classes such as CP-theories representing CP-nets or TCP-nets, w is bounded by a constant, and so the complexity is then $O(mn)$.

5 Comparison and Discussion

An upper approximation $\succ_{p(\Gamma)}$ for $\succ_\Gamma$ was defined in [13], which was refined to upper approximation $\gg_\Gamma$ in [12]. It follows easily from their construction that $\gg_\Gamma \subseteq \succ_{p(\Gamma)}$ when the latter is defined.

Both these require consistency, and strong acyclicity properties on the ordering of variables used in statements in Γ. In particular, in both cases, it must be possible to label the set of variables V as $\{X_1, \ldots, X_n\}$ in such a way that for any $\varphi \in \Gamma$, if $X_i \in U_\varphi$, and $X_j \in \{X_\varphi\} \cup W_\varphi$ then $i < j$. It can be proved using Proposition 5 that $\trianglerighteq$ is never a worse upper approximation than $\gg_\Gamma$ or $\succ_{p(\Gamma)}$.

Proposition 6 *Let α and β be two different outcomes such that $\alpha \trianglerighteq \beta$. Then $\alpha \gg_\Gamma \beta$ if Γ is such that $\gg_\Gamma$ is defined; hence also $\alpha \succ_{p(\Gamma)} \beta$ if $\succ_{p(\Gamma)}$ is defined.*

Example continued. The upper approximations of [13, 12] are not applicable because Γ is not consistent. If we restore consistency by removing φ_4 from Γ then they *are* applicable but give a poorer upper approximation; in particular they both have $\mathrm{x y \bar{z}}$ preferred to $\mathrm{x \bar{y} z}$ (essentially because they make Y a more important variable than Z, as Y is a parent of Z), so that they miss undominated solution $\mathrm{x \bar{y} z}$.

Application to constrained optimisation

In the constrained optimisation algorithm in [3], and the amendments in [13, 12], a search tree is used to find solutions, where the search tree is chosen so that its associated total ordering on outcomes extends $\succ_\Gamma$. Methods for finding such search trees have been developed in [12]. We can make use of this search tree as follows, amending the constrained optimisation approach of [12] in the obvious way: when we find a new solution α we check if it is $\trianglerighteq$-undominated with respect to each of the current known set K of $\trianglerighteq$-undominated solutions. If so, then α is an $\trianglerighteq$-undominated solution, since it cannot be $\trianglerighteq$-dominated by any solution found later. We add α to K, and continue the search. This is an anytime algorithm, but if we continue until the end of the search, K will be the complete set of $\trianglerighteq$-undominated solutions, which is a subset of the set of $\succ_\Gamma$-undominated solutions, since $\trianglerighteq \supseteq \succ_\Gamma$. For the inconsistent case, by definition, no such search tree can exist. Instead we could use a pre-ordered search tree satisfying Γ, and continue generating solutions with this for as long as it totally orders solutions. This may well be successful for cases where the inconsistency is relatively localised and among less preferred outcomes, such as in the example.

Crudeness of the approximation

As illustrated by the example, $\trianglerighteq_\Gamma$ can be a close approximation of $\succ_\Gamma$. However, computing dominance with respect to $\succ_\Gamma$ appears in general to be extremely hard [7] whereas our approximation $\trianglerighteq_\Gamma$ is of low order polynomial complexity. One would therefore not expect $\trianglerighteq_\Gamma$ always to be a close approximation. To illustrate this, consider outcomes α and β which differ on all variables (or, more generally on all variables not in $W_\Gamma = \bigcup_{\varphi\in\Gamma} W_\varphi$). Then any variable which is pickable is decisive, so, by Proposition 5, $\beta \trianglerighteq_\Gamma \alpha$ if and only if there exists no variable which is pickable given $\top$ with respect to (α, β). A variable is $\top$-choosable if and only if it is not in W_Γ, so $\beta \trianglerighteq_\Gamma \alpha$ if and only if for all $Y \in V - W_\Gamma$, $\beta(Y) \sqsupset_\top^Y \alpha(Y)$. The relation $\sqsupset_\top^Y$ does not depend at all on u_φ, for $\varphi \in \Gamma$, so, for such α and β, whether $\beta \trianglerighteq_\Gamma \alpha$ holds or not does not depend at all on u_φ, for $\varphi \in \Gamma$. In particular, if $\beta \succ_{\Gamma_*} \alpha$ holds, where Γ_* is Γ in which each u_φ is changed to $\top$, then, by Proposition 1, $\beta \trianglerighteq_\Gamma \alpha$ holds, since for such α and β, $\beta \trianglerighteq_\Gamma \alpha$ if and only if $\beta \trianglerighteq_{\Gamma_*} \alpha$. Since $\succ_{\Gamma_*}$ can easily be a very crude upper approximation of $\succ_\Gamma$, this suggests that $\trianglerighteq_\Gamma$ may often not be a close approximation for such pairs of outcomes, i.e., we may easily have $\beta \trianglerighteq_\Gamma \alpha$ without $\beta \succ_\Gamma \alpha$.

However, this does not necessarily matter for constrained optimisation. There will often be a very large number of optimal solutions, and we may well only wish to report a small fraction of them; it is not necessarily important that the upper approximation is a close approximation, just that $\trianglerighteq$ is a sufficiently sparse (i.e., weak) relation, so that there are still liable to be a good number of solutions which are $\trianglerighteq$-undominated.

Summary

In this paper we have constructed a new upper approximation of conditional preference in CP-theories, which is very much more widely applicable than previous approaches, as well as being a better approximation. Furthermore, an efficient algorithm for dominance testing with respect to approximate preference has been derived.

ACKNOWLEDGEMENTS

This material is based upon works supported by the Science Foundation Ireland under Grant No. 00/PI.1/C075. I'm grateful to the referees for their helpful comments.

REFERENCES

[1] C. Boutilier, R. Brafman, H. Hoos, and D. Poole, 'Reasoning with conditional *ceteris paribus* preference statements', in *Proceedings of UAI-99*, pp. 71–80, (1999).

[2] C. Boutilier, R. I. Brafman, C. Domshlak, H. Hoos, and D. Poole, 'CP-nets: A tool for reasoning with conditional *ceteris paribus* preference statements', *Journal of Artificial Intelligence Research*, **21**, 135–191, (2004).

[3] C. Boutilier, R. I. Brafman, C. Domshlak, H. Hoos, and D. Poole, 'Preference-based constrained optimization with CP-nets', *Computational Intelligence*, **20**(2), 137–157, (2004).

[4] R. Brafman and C. Domshlak, 'Introducing variable importance tradeoffs into CP-nets', in *Proceedings of UAI-02*, pp. 69–76, (2002).

[5] R. I. Brafman and Y. Dimopoulos, 'Extended semantics and optimization algorithms for CP-networks', *Computational Intelligence*, **20**(2), 218–245, (2004).

[6] C. Domshlak and R. I. Brafman, 'CP-nets—reasoning and consistency testing', in *Proc. KR02*, pp. 121–132, (2002).

[7] J. Goldsmith, J. Lang, M. Truszczyński, and N. Wilson, 'The computational complexity of dominance and consistency in CP-nets', in *Proceedings of IJCAI-05*, pp. 144 –149, (2005).

[8] J. Lang, 'Logical preference representation and combinatorial vote', *Ann. Mathematics and Artificial Intelligence*, **42**(1), 37–71, (2004).

[9] M. McGeachie and J. Doyle, 'Utility functions for ceteris paribus preferences', *Computational Intelligence*, **20**(2), 158–217, (2004).

[10] S. Prestwich, F. Rossi, K. B. Venable, and T. Walsh, 'Constrained CP-nets', in *Proceedings of CSCLP'04*, (2004).

[11] S. Prestwich, F. Rossi, K. B. Venable, and T. Walsh, 'Constraint-based preferential optimization', in *Proceedings of AAAI-05*, (2005).

[12] N. Wilson, 'Consistency and constrained optimisation for conditional preferences', in *Proceedings of ECAI-04*, pp. 888–892, (2004).

[13] N. Wilson, 'Extending CP-nets with stronger conditional preference statements', in *Proceedings of AAAI-04*, pp. 735–741, (2004).