Efficient Inference for Expressive Comparative Preference Languages

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

Abstract

A fundamental task for reasoning with preferences is the following: given input preference information from a user, and outcomes α and β, should we infer that the user will prefer α to β? For CP-nets and related comparative preference formalisms, inferring a preference of α over β using the standard definition of derived preference appears to be extremely hard, and has been proved to be PSPACE-complete in general for CP-nets. Such inference is also rather conservative, only making the assumption of transitivity. This paper defines a less conservative approach to inference which can be applied for very general forms of input. It is shown to be efficient for expressive comparative preference languages, allowing comparisons between arbitrary partial tuples (including complete assignments), and with the preferences being *ceteris paribus* or not.

1 INTRODUCTION

Recent years have seen a considerable literature develop in the Artificial Intelligence community on formalisms for reasoning with comparative preferences in combinatorial problems, involving statements which compactly express the relative preference of outcomes (complete assignments to a set of variables). For example, consider a CP-net [Boutilier *et al.*, 1999; 2004] statement φ of the form $u : x > x'$ where x and x' are assignments to a single variable X, and u is an assignment to some set of other variables U. This compactly represents the typically exponentially large set φ^* of all pairs (α, β) of outcomes such that (i) α and β both extend u, (ii) $\alpha(X) = x$, $\beta(X) = x'$ and (iii) α and β agree on all other variables. Each pair $(\alpha, \beta) \in \varphi^*$ represents a preference for α over β.

CP-nets and related formalisms [Boutilier *et al.*, 2004; Brafman *et al.*, 2006; Wilson, 2004a] allow only very restricted forms of comparative preference statements. We can consider much more general statements φ, with associated relation φ^* giving the preferences between alternatives that are directly implied by φ. Each pair (α, β) in φ^* represents a preference (e.g., of a single user) for outcome α over outcome β. Many comparative preference statements may be elicited, to form a set Γ. This set Γ thus directly represents preferences $\Gamma^* = \bigcup_{\varphi \in \Gamma} \varphi^*$. A key problem is the following:

Given that the user has told us Γ is it reasonable to infer that the user prefers outcome α to outcome β?

We take a semantic approach to inference, related to that taken in [Boutilier *et al.*, 2004; Brafman and Dimopoulos, 2004; Wilson, 2004b; 2006]. Let us assume that the user's preference relation[1] $\succcurlyeq$ is a **weak order**, also known as a *total pre-order*, i.e., a relation on outcomes which is transitive and complete, so that (i) $\alpha \succcurlyeq \beta$ and $\beta \succcurlyeq \gamma$ implies $\alpha \succcurlyeq \gamma$, and (ii) for all outcomes α and β, either $\alpha \succcurlyeq \beta$ or $\beta \succcurlyeq \alpha$. A weak order *satisfies* a comparative preference statement φ if and only if it extends φ^*. We write $\Gamma \models \varphi$ (and also $\Gamma^* \models \varphi^*$) if and only if every weak order satisfying every statement in Γ also satisfies φ. It can be shown that this holds if and only if φ^* is a subset of the transitive closure of Γ^*.

Unfortunately, even for quite restrictive languages, it appears to be extremely hard in general to test if Γ entails, in this sense, a preference of α over β, i.e., if $\Gamma^* \models \{(\alpha, \beta)\}$. In particular, for CP-nets such dominance testing has been shown to be PSPACE-complete [Goldsmith *et al.*, 2005]. In addition, this kind of inference is conservative, and tends to lead to weak preferences with a great deal of incomparability. It is very often desirable to be able to fill out the user's direct preferences in a plausible way by some kind of extrapolation, generating a fuller relation. This can be done by basing the inference relation on a smaller set of weak orders.

Note, however, that the aim is not to generate a single utility function (or weak order) compatible with the inputs (cf. the approaches in [Boutilier *et al.*, 2001; Domshlak *et al.*, 2003; McGeachie and Doyle, 2004; Brafman and Domshlak, 2008]). This would need to involve many arbitrary choices, since the output preferences are so much stronger than the inputs, and therefore the utility function would be extremely unlikely to be compatible with the user's unknown preferences.

Our approach involves defining entailment using only weak orders which are of a kind of generalised lexicographic

[1] Equivalently we might assume that the user's preference is represented by a utility function which assigns a real value to each outcome, with $U(\alpha) \geq U(\beta)$ indicating that the user prefers α to β. The focus on weak orders, as in [Brafman and Dimopoulos, 2004; Wilson, 2006], rather than on total orders, means that the approach also applies to situations where the preferences can imply cycles of outcomes.

form. It extends and generalises the upper approximation described in [Wilson, 2006], especially, in allowing very much more general comparative preference input statements. In fact, the approach applies to completely arbitrary input preference relations, and the inference algorithm is polynomial for many natural kinds of preference statements, such as comparisons between arbitrary partial tuples (including complete assignments), and with the preferences being *ceteris paribus* or not. No acyclicity conditions are required regarding the input statements, and consistency (i.e., acyclicity of the preference relation) is not assumed.

Section 2 describes some different forms of comparative preference statements, and an operation on them which is important for our computational approach. Section 3 describes a special kind of weak order on outcomes, those generated by what we call a *cp-tree*. In the semantics, restricting to this kind of weak order leads to a stronger form of inference, as defined in Section 4. Section 5 describes our computational technique for determining entailment, which is polynomial for a wide range of comparative preference statements. Section 6 sketches the proof of correctness of the technique in Section 5. Section 7 discusses application to a broader range of comparative preference statements.

Terminology. 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 $A \subseteq V$ let $\underline{A} = \prod_{X \in A} \underline{X}$ be the set of possible assignments to set of variables A. 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. If $a \in \underline{A}$ is an assignment to A, and $b \in \underline{B}$, where $A \cap B = \emptyset$, then we may write ab as the assignment to $A \cup B$ which combines a and b. 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 $\alpha(U) = u$. This is if and only if u and a *agree* on common variables, i.e., $u(A \cap U) = a(A \cap U)$. Otherwise, we say that a and u are *incompatible*.

Let $\succcurlyeq$ be some transitive relation on a set Z. We say that z_1 and z_2 are *$\succcurlyeq$-equivalent* if both $z_1 \succcurlyeq z_2$ and $z_2 \succcurlyeq z_1$.

2 COMPARATIVE PREFERENCE STATEMENTS

In this paper we will focus especially on comparative preference statements φ of the form $p > q \parallel T$, where P, Q and T are subsets of V, and $p \in \underline{P}$ is an assignment to P, and $q \in \underline{Q}$ is an assignment to Q. Informally, the statement $p > q \parallel T$ represents the following: p is preferred to q if T is held constant. Formally, the semantics of this statement is given by the relation φ^* which is defined to be the set of pairs (α, β) of outcomes such that α extends p, and β extends q, and α and β agree on T: $\alpha(T) = \beta(T)$.

If p and q do not agree on common variables in T, i.e., if $p(P \cap Q \cap T) \neq q(P \cap Q \cap T)$, then φ is vacuous: $\varphi^* = \emptyset$. Given that p and q do agree on common variables in T, we can assume, without loss of generality, that $P \cap T = \emptyset$ and $Q \cap T = \emptyset$, since, for example, we can add any variable X in $(P \cap T) - Q$ to Q, extending q by $q(X) = p(X)$, and remove X from T, without changing φ^*.

We are particularly interested in such statements φ when $P = Q$. The statement can then be written as $us > us' \parallel T$, where U, S and T are disjoint sets of variables, and $u \in \underline{U}$, and s and s' are assignments to S which differ on each variable: $s(Z) \neq s'(Z)$ for all $Z \in S$.

Ceteris paribus preferences are represented by statements with $T = V - (U \cup S)$; a *feature vector rule* in [McGeachie and Doyle, 2004] can be represented by such a statement. On the other hand, statements with $T = \emptyset$ represent a stronger kind of preference, which can be used, for example, for representing lexicographic and similar orders. A CP-theory [Wilson, 2004b; 2004a] statement $u : x > x'$ $[W]$ is exactly equivalent to statement $us > us' \parallel T$ when we set $S = \{X\}$, $x = s$, $x' = s'$ and $T = V - (U \cup \{X\} \cup W)$. CP-nets [Boutilier *et al.*, 1999; 2004] and TCP-nets [Brafman and Domshlak, 2002; Brafman *et al.*, 2006] can be expressed in terms of CP-theories [Wilson, 2004a]. In particular, CP-net statement $u : x > x'$ is equivalent to a CP-theory statement when $W = \emptyset$ and so $T = V - (U \cup \{X\})$. A preference of outcome α over outcome β can be expressed by a statement $us > us' \parallel \emptyset$ by setting $u = \alpha(U) = \beta(U)$, where U is the variables on which α and β agree, and $s = \alpha(V - U)$ and $s' = \beta(V - U)$. A more general statement $p > q \parallel T$ can also be used to represent statements of the form "α is the best outcome extending tuple u".

Selection-projections. The computational technique described in this paper is efficient essentially if and only if one can efficiently compute a particular compound operation on the input comparative preference statements: the projection of a selection. Fortunately, this operation is efficient for a broad class of natural comparative preference statements. Let a be an assignment to set of variables A, and let Y be a set of variables disjoint with A. For relation R on the set of outcomes, define the a-selection R_a of R to consist of all pairs (α, β) in R such that both α and β extend a. We define, for $Y \subseteq V$, the projection $R^{\downarrow Y}$ of R to be the set of all pairs $(y, y') \in \underline{Y} \times \underline{Y}$ such that there exists tuples z and z' with $(yz, y'z') \in R$. We write R_a^Y for $(R_a)^{\downarrow Y}$, the projection to Y of the a-selection of R. We call this compound operation a *selection-projection*. Let $y, y' \in \underline{Y}$ be assignments to Y. We have $(y, y') \in R_a^Y$ if and only if there exist assignments z, z' to $V - (A \cup Y)$ such that $(ayz, ay'z') \in R$.

The following property, which is important for the efficiency of our approach, follows easily from the definitions.

Proposition 1 (Decomposition) *For i in some index set I, let R_i be some relation on outcomes, and let $R = \bigcup_{i \in I} R_i$. Let a be an assignment to set of variables A, and let Y be a set of variables disjoint with A. Then $R_a^Y = \bigcup_{i \in I} (R_i)_a^Y$.*

For comparative preference statement φ and set of comparative preference statements Γ we abbreviate $(\varphi^*)_a^Y$ to φ_a^Y and abbreviate $(\Gamma^*)_a^Y$ to Γ_a^Y. We thus have $\Gamma_a^Y = \bigcup_{\varphi \in \Gamma} \varphi_a^Y$. We are interested in sets Y whose associated product set $\underline{Y}$ is not large (so, small sets of variables whose domains are fairly small). Then the relations Γ_a^Y are of manageable size, even though Γ^* may very well be exponentially large.

Proposition 2 *Let P, Q and T be subsets of V, and let $p \in \underline{P}$ be an assignment to P, and $q \in \underline{Q}$ be an assignment to Q. Let φ be a comparative preference statement of the form $p > q \parallel T$, as defined above, where $p \in \underline{P}$, $q \in \underline{Q}$ and $(P \cup Q) \cap T = \emptyset$. Let a be an assignment to a set of variables A, and let Y be a set of variables disjoint from A. φ_a^Y is empty unless a is compatible with both p and q. If a is compatible with both p and q then φ_a^Y consists of all pairs (y, y') such that (i) y and y' agree on $Y \cap T$, i.e., $y(Y \cap T) = y'(Y \cap T)$; (ii) y is compatible with p; and (iii) y' is compatible with q.*

Each of these conditions can be checked in time at worst linear in n, the number of variables, and so the relation φ_a^Y can be computed in time linear in n, given that the size of $\underline{Y}$ is bounded by a constant. Proposition 2 therefore shows that computing selection-projection can be achieved efficiently for statements of the form $p > q \parallel T$, and hence, by Proposition 1, for sets Γ of such statements.

For compatible p and q a variation of the above definition of φ^* will often be natural, which insists in addition that, for $(\alpha, \beta) \in \varphi^*$, α does not extend q, and β does not extend p (cf. [McGeachie and Doyle, 2004]). It can be shown that φ_a^Y can be computed efficiently in this case also.

3 cp-TREES AND THEIR WEAK ORDERS

In this section we define a special kind of weak order, one generated by what we call a *cp-tree*. They are generalisations of the pre-ordered search trees of [Wilson, 2006]. cp-trees represent a rather natural and simple model of a user's preferences. In such a model, the user orders outcomes as follows: they first choose a small set of variables Y and an ordering $\geqslant$ on the assignments to Y. Outcomes are primarily ordered by considering their projections to Y, comparing them using $\geqslant$. If the outcomes do not differ on Y then a disjoint set of variables Y' is considered next, along with an ordering $\geqslant'$ on $\underline{Y'}$, and so on. cp-trees therefore represent a form of lexicographic order, but where the importance ordering on variables can depend on more important variables, as can the value orderings.

A cp-tree is a rooted directed tree, which we picture being drawn with the root at the top, and children below parents. Associated with each node r in the tree is a set of variables Y_r, which is instantiated with a different assignment in each of the node's children (if it has any), and also a weak order $\geqslant_r$ of the values of Y_r.

More formally, define a **cp-node** r (usually abbreviated to just "node") 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, $Y_r \subseteq V - A_r$ is a non-empty set of other variables; $\geqslant_r$ is a weak order on the set $\underline{Y_r}$ of values of Y_r which is not equal to the trivial full relation on $\underline{Y}$; so there exists some $y, y' \in \underline{Y}$ with $y \not\geqslant_r y'$.

A **cp-tree** is defined to be a directed tree, where edges are directed away from a root node, root, so that all nodes apart from the root node have a unique parent node. The *ancestors* of a node r are the nodes on the path from root to the parent node of r. Each node is identified with a unique cp-node r. Let $r \to r'$ be an edge in the cp-tree from a node r to one of its children r'. Associated with this edge is an assignment y to variables Y_r. This is different from the assignment y' associated with any other edges from node r. $A_{r'} = A_r \cup Y_{r'}$, and $a_{r'}$ is a_r extended with the assignment $Y_r = y$. We also have $A_{\mathrm{root}} = \emptyset$. Therefore A_r is the union of sets $Y_{r''}$ over all ancestors r'' of r; and a_r consists of all assignments made on the path from the root to r. The root node has $a_{\mathrm{root}} = \top$, the assignment to the empty set of variables.

We also assume that the weak orders $\geqslant_r$ satisfy the following condition, (to ensure 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 assignments (of Y_r) associated with the children of r. The only difference between a cp-tree and a pre-ordered search tree as defined in [Wilson, 2006] is that, in the latter, Y_r is just a single variable, rather than a non-empty set of variables.

The weak order $\succcurlyeq_\sigma$ associated with a cp-tree σ

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.

Definition 1 *Let σ be a cp-tree. The associated relation $\succcurlyeq_\sigma$ on outcomes is defined as follows: For outcomes $\alpha, \beta \in \underline{V}$ 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 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 complete; it is easily shown to be transitive, and is hence a weak order.

We say that cp-tree σ ***satisfies*** *relation* R if and only if $\succcurlyeq_\sigma$ satisfies R i.e., $\succcurlyeq_\sigma$ extends R, that is, $(\alpha, \beta) \in R \Rightarrow \alpha \succcurlyeq_\sigma \beta$. Similarly, for comparative preference statement φ and set of comparative statements Γ, we say that σ *satisfies* φ (respectively, Γ) if and only if σ satisfies φ^* (respectively, Γ^*).

Example. I am planning a holiday, and I have to decide where, when, and for how long I want to go, represented by variables X_1 (Paris or London), X_2 (Spring or Summer) and X_3 (one week or two weeks), respectively. Let V be the set of variables $\{X_1, X_2, X_3\}$ with domains $\underline{X_i} = \{x_i, x_i'\}$, for $i = 1, 2, 3$, where x_1 represents Paris, x_2 represents Spring,

x_3 represents going for one week, etc. Define a cp-tree σ with two nodes, root $= \langle \emptyset, \top, \{X_1\}, [x_1 > x_1'] \rangle$ and its only child node $r = \langle \{X_1\}, x_1, \{X_3\}, [x_3 > x_3'] \rangle$. Let φ_1 be the comparative preference statement $x_1 x_2 > x_1' \parallel \{X_3\}$, which represents that I'd rather go to Paris in the Springtime than London any time. The associated set of pairs of outcomes φ_1^* consists of $(x_1x_2x_3, x_1'x_2, x_3)$, $(x_1x_2x_3', x_1'x_2, x_3')$, $(x_1x_2x_3, x_1'x_2', x_3)$, and $(x_1x_2x_3', x_1'x_2', x_3')$.

Each of these is in $\succcurlyeq_\sigma$, which implies that $\succcurlyeq_\sigma$ extends φ_1^*, and so cp-tree σ satisfies φ_1. Similarly, σ satisfies φ_2 and φ_3, where $\varphi_2 = x_2'x_3 > x_2x_3' \parallel \{X_1\}$, and $\varphi_3 = x_1'x_3' > x_1'x_3 \parallel \emptyset$. Node r decides outcomes $x_1x_2x_3'$ and $x_1x_2x_3$, and we have $x_1x_2x_3 >_r x_1x_2x_3'$, and so $x_1x_2x_3 \succ_\sigma x_1x_2x_3'$, i.e., $x_1x_2x_3' \not\succcurlyeq_\sigma x_1x_2x_3$. This implies that σ fails to satisfy the statement ψ equalling $x_1x_2x_3' > x_1x_2x_3$, since $\succcurlyeq_\sigma$ does not extend $\psi^* = \{(x_1x_2x_3', x_1x_2x_3)\}$.

A cp-tree ordering is a special kind of weak order in which there is a most important set Y' of variables, and outcomes are compared first on these. Only if the outcomes agree on Y' are further variables considered. If someone had the same preferences as cp-tree σ in the Example, then they would regard that X_1 (destination) is the (uniquely) most important variable, with x_1 (Paris) being better than x_1' (London). Given x_1, X_3 (length of stay) is the next most important variable, with x_3 being better than x_3'. cp-tree orderings thus represent quite a simple way of ordering outcomes, but one which seems fairly psychologically plausible (people often focus first on a small set of variables).

4 DEDUCTION BASED ON cp-TREES

We fix a family $\mathcal{Y}$ of small subsets of V. For example, $\mathcal{Y}$ might be defined to be all singleton subsets of V (i.e., sets with cardinality of one), or, alternatively, all subsets of cardinality at most two, and so on.

Definition 2 ($\mathcal{Y}$-cp-tree) *Let $\mathcal{Y}$ be a set of non-empty subsets of V such that if $Y \in \mathcal{Y}$ and non-empty Y' is a subset of Y then $Y' \in \mathcal{Y}$. A $\mathcal{Y}$-cp-tree is defined to be a cp-tree σ such that for any node r of σ, we have $Y_r \in \mathcal{Y}$.*

$\mathcal{Y}$-entailment $\models_\mathcal{Y}$. We can now consider deduction of comparative preferences based on $\mathcal{Y}$-cp-trees. Let R be a relation on outcomes, and let ψ be a comparative preference statement. $R \models_\mathcal{Y} \psi$ holds if and only if every $\mathcal{Y}$-cp-tree satisfying R also satisfies ψ^*. Let Γ be a set of comparative preference statements. $\Gamma \models_\mathcal{Y} \psi$ holds if and only if $\Gamma^* \models_\mathcal{Y} \psi^*$. In the example, using $\mathcal{Y} = \{\{X_1\}, \{X_2\}, \{X_3\}\}$, we have $\{\varphi_1, \varphi_2, \varphi_3\} \not\models_\mathcal{Y} \psi$ because there exists a $\mathcal{Y}$-cp-tree satisfying $\{\varphi_1, \varphi_2, \varphi_3\}$ but not ψ.

An especially important case is when ψ^* is just equal to a singleton set $\{(\alpha, \beta)\}$ for some outcomes α and β. Such an inference from Γ is an inferred preference for α over β. For input relation R on outcomes we define the inferred preference relation $R_\mathcal{Y}$ to consist of all pairs (α, β) such that $R \models_\mathcal{Y} \{(\alpha, \beta)\}$. It is clear that $R_\mathcal{Y}$ contains R. It is also transitive, since it is the intersection of a set of transitive relations (the set of $\succcurlyeq_\sigma$ over all σ satisfying R). It hence contains the transitive closure of R. Therefore, for the special cases of CP-theories and CP-nets, $R_\mathcal{Y}$ contains the standard preference relation: it is an upper approximation [Wilson, 2006]. In particular, we associate preference relation $\Gamma_\mathcal{Y}^*$ with set Γ of comparative preference statements (in some language). If we let $\mathcal{Y}$ be the set of singleton subsets of V, and consider only comparative preference statements of the form $ux > ux' \parallel T$ where x and x' are different values of a variable X (see Section 2) then $\Gamma_\mathcal{Y}^*$ is the same as the preference relation $\trianglerighteq_\Gamma$ defined in [Wilson, 2006].

We will consider somewhat more general preference statements ψ, of the following form: all outcomes extending p are preferred to all outcomes extending p', where p and p' are given assignments to set of variables $P \subseteq V$. We can write this, using the notation developed in Section 2, as a statement $us > us' \parallel \emptyset$.

5 COMPUTATION OF $\mathcal{Y}$-ENTAILMENT

This section describes an algorithm for $\mathcal{Y}$-entailment, which is polynomial for a wide range of comparative preference relations; in particular, statements of the form $p > q \parallel T$ as described in Section 2, or any other comparative statements for which computing selection-projections is polynomial.

The approach very substantially generalises that in [Wilson, 2006]. A completely arbitrary input relation R is allowed (which simplifies some of the results). We also allow a much richer language of output queries ψ, and a richer class of models, by allowing elements of $\mathcal{Y}$ to be non-singleton sets; these two things complicate some proofs; however, the proof follows the same structure as that in [Wilson, 2006].

Throughout this section, we consider a fixed family $\mathcal{Y}$ of sets of variables, which parameterises the inference relation, and a fixed input relation R on outcomes. We also consider a fixed comparative preference statement ψ of the form $us > us' \parallel \emptyset$, as defined in Section 4. U and S are disjoint sets of variables, and $u \in \underline{U}$, and s and s' are assignments to S which differ on each variable in S.

Definition 3 (Pickable and Decisive) *Given set $Y \subseteq V$ and assignment a to some subset A of $V - Y$, we define $\sqsupseteq_a^Y$ to be the transitive closure of R_a^Y. Suppose that u is compatible with $a \in \underline{A}$. Set of variables Y is said to be ψ-***pickable** *given a if $Y \cap A = \emptyset$ and either*

(i) $Y \subseteq U$ and $u(Y)$ is not $\sqsupseteq_a^Y$-equivalent to any other assignment in $\underline{Y}$; or

(ii) $Y \not\subseteq U$ and there exists $y, y' \in \underline{Y}$ with $y \not\sqsupseteq_a^Y y'$ and y is compatible with us, and y' is compatible with us'.

*In case (ii) we say that Y is ψ-***decisive** *given a.*

Algorithm for determining $\mathcal{Y}$-entailment

The following algorithm assumes a given input relation R on outcomes and comparative preference statement ψ of the form $us > us' \parallel \emptyset$. It determines if R $\mathcal{Y}$-entails ψ or not, i.e., if $R \models_\mathcal{Y} \psi^*$, where family of subsets $\mathcal{Y}$ is as defined in Section 4, parameterises the deduction relation.

procedure Does R $\mathcal{Y}$-entail ψ?

for $j := 1, \ldots, n$

let a_j be u restricted to $Y_1 \cup \cdots \cup Y_{j-1}$ (in particular, $a_1 = \top$);

if there exists a set in $\mathcal{Y}$ which is ψ-decisive given a_j
  then return **false** and **stop**;
if there exists a set in $\mathcal{Y}$ which is ψ-pickable given a_j
  then let Y_j be any such set;
  else return **true** and **stop**;
next j;
return **true**.

The theorem states the correctness of the algorithm. Section 6 shows how this is proved.

Theorem 1 *Let R be a relation on outcomes, and let ψ be a comparative preference statement of the form $us > us' \parallel \emptyset$. The above procedure is correct, i.e., it returns **true** if $R \models_{\mathcal{Y}} \psi^*$ and it returns **false** if $R \not\models_{\mathcal{Y}} \psi^*$.*

Application to Deduction for Comparative Preference Statements

The algorithm applies to arbitrary input relations R. Relation R will very often be exponentially large, and so will need to represented compactly, in particular as a set Γ of comparative preference statements (in some language), where Γ represents relation $R = \Gamma^*$ on outcomes. We infer ψ of the form $us > us' \parallel \emptyset$ from Γ if and only if $\Gamma^* \models_{\mathcal{Y}} \psi^*$. Applying the approach described above requires us to compute selection-projections of the form Γ_a^Y, which we can compute as $\bigcup_{\varphi \in \Gamma} \varphi_a^Y$ using Proposition 1.

Example continued. We can write ψ as $us > us'$, where $U = \{X_1, X_2\}$, and $u = x_1 x_2$, and $s = x_3'$ and $s' = x_3$. We wish to determine if we can infer ψ, representing a preference for Paris-Spring-TwoWeeks over Paris-Spring-OneWeek. $\{X_1\}$ is ψ-pickable because $\{X_1\}$ is a subset of U (case (i)), and $u(X_1)$, i.e., x_1, is not equivalent to x_1' (since none of the three statements in Γ give a preference of x_1' over x_1). $\{X_3\}$ is then ψ-decisive given x_1 since:- X_3 is not in U so we're in case (ii); $\Gamma_{x_1}^{X_3}$ does not contain the pair (x_3', x_3), since none of the statements in Γ give a preference for x_3' over x_3 in a context compatible with x_1. (φ_3 gives a preference for x_3' over x_3 but only when x_1' holds, which is incompatible with x_1, and so $(\varphi_3)_{x_1}^{X_3}$ is empty.) So the algorithm returns **false**. This implies (see Section 6) that there exists a cp-tree (e.g., σ defined in Section 3) which satisfies Γ but does not satisfy ψ. Hence Γ does not $\mathcal{Y}$-entail ψ.

On the other hand, consider the statement $\psi' = x_1 x_2 x_3' > x_1' x_2 x_3$. None of the variables are ψ'-pickable. For example, X_1 is not ψ'-pickable because $(\varphi_1)_{\top}^{X_1}$ contains (x_1, x_1'). Hence the algorithm returns **true**, and so Γ $\mathcal{Y}$-entails ψ'.

Complexity. Let $\mathcal{L}$ be the set of comparative preferences which can be written in the form $p > q \parallel T$ (see Section 2). This includes all CP-net statements and CP-theory statements, along with more complex comparisons between tuples. Assume that $\Gamma \subseteq \mathcal{L}$. Computing Γ_a^Y can then be performed efficiently, using Propositions 1 and 2. Assume that the domain sizes are bounded above by a constant, and that the elements of $\mathcal{Y}$ have cardinality at most k, and so $|\mathcal{Y}|$ is less than n^k. The algorithm is then $O(mn^{k+1})$, where $m = |\Gamma|$, (using the fact that, for a statement $p > q \parallel T$ in Γ, checking the compatibility of a with p and q can be performed incrementally).

6 PROVING THE THEOREM

The main aim of this section is to sketch how to prove Theorem 1, which states the correctness of the algorithm for $\mathcal{Y}$-entailment in Section 5. (A longer version of the paper including proofs can be downloaded from the author's website.) We are addressing the problem of whether relation R $\mathcal{Y}$-entails ψ, where comparative preference statement ψ is of the form $us > us' \parallel \emptyset$. By definition, this fails to hold if and only if there exists a $\mathcal{Y}$-cp-tree which satisfies R but not ψ. Proposition 3 shows that such cp-trees map to particular sequences of sets in $\mathcal{Y}$, which we call *decisive sequences*. $\mathcal{Y}$-entailment of ψ from R can therefore be determined by checking for the existence of a decisive sequence. A decisive sequence is a sequence of *pickable* $Y \in \mathcal{Y}$ ending with a *decisive* Y. The pickable Y correspond to nodes in a cp-tree satisfying R, and a decisive Y corresponds to a node in cp-tree which satisfies R but not ψ. Determining if a Y is pickable or decisive can be done using a selection-projection of R, so is efficient if and only if this selection-projection can be done efficiently.

A monotonicity property, expressed by Proposition 4, states, roughly speaking, that a set Y which is pickable remains pickable if other sets are chosen first. This means that the search for decisive sequences can be performed in a backtrack-free manner, leading to the simple and efficient algorithm below whose correctness is stated by the theorem. The algorithm iteratively checks to see if there exists a decisive set Y, and if so, it returns "false", meaning R does not $\mathcal{Y}$-entail ψ, since a decisive sequence has been constructed. If not, it checks if there is a pickable set Y. If there is no such set Y then it returns "true", as there is then no decisive sequence.

We wish to determine if relation R $\mathcal{Y}$-entails ψ, where comparative preference statement ψ is of the form $us > us' \parallel \emptyset$. We map cp-trees, which satisfy R but not ψ, to a sequence of sets $Y \in \mathcal{Y}$, which we call a *decisive sequence*, generated from the cp-nodes on the path from the root to a node which falsifies ψ (i.e., the node's ordering is incompatible with ψ).

A ($\mathcal{Y}$-)***decisive sequence*** *(w.r.t. ψ)* is defined to be a sequence $Y_1, \ldots, Y_k$ of disjoint sets in $\mathcal{Y}$ satisfying the following conditions:

— for $j = 1, \ldots, k-1$, $Y_j \subseteq U$, and $Y_k \not\subseteq U$;

— for $j = 1, \ldots, k$, Y_j is ψ-pickable given a_j where a_j is u restricted to $Y_1 \cup \cdots \cup Y_{j-1}$; in particular, Y_k is decisive given a_k.

The following result shows that entailment is equivalent to the absence of a decisive sequence. A cp-tree which is in the form of a chain is generated from a decisive sequence, where a set Y in the sequence generates a node r with $Y_r = Y$. Conversely, the decisive sequence is generated from a particular path in the cp-tree.

Proposition 3 *There exists a $\mathcal{Y}$-decisive sequence w.r.t. ψ if and only if it is not the case that $R \models_{\mathcal{Y}} \psi$, i.e., if and only if* *there exists $(\alpha, \beta) \in \psi^*$ and a $\mathcal{Y}$-cp-tree σ satisfying R with $\beta \succ_\sigma \alpha$.*

Theorem 1 follows easily using Proposition 3 and the following monotonicity result.

Proposition 4 *Let ψ be a comparative preference statement of the form $us > us' \parallel \emptyset$, where $u \in \underline{U}$. Suppose that $A \subseteq B \subseteq U$, and that $a \in \underline{A}$ and $b \in \underline{B}$ and that u extends b which extends a. Let $Y \subseteq V$ be such that $A \cap Y = \emptyset$ and that $Y \not\subseteq B$.*

If $Y \in \mathcal{Y}$ is ψ-pickable given a then $Y - B$ is ψ-pickable given b. If $Y \in \mathcal{Y}$ is ψ-decisive given a then $Y - B$ is ψ-decisive given b.

7 SUMMARY AND DISCUSSION

We have defined a form of inference for comparative preferences, which we call $\mathcal{Y}$-entailment, where $\mathcal{Y}$ is a set of small non-empty sets of variables which parameterises the inference. For example, $\mathcal{Y}$ might the the set of singleton subsets; or alternatively, the set of all non-empty subsets with at most two elements. We have given a sound and complete algorithm for determining if set of comparative preference statements Γ $\mathcal{Y}$-entails ψ where ψ is a comparative preference statement of a special form; in particular, ψ can represent a preference for an outcome α over an outcome β. As described in [Wilson, 2006], we can use $\mathcal{Y}$-entailment in a constrained optimisation algorithm. Being able to deduce a more general comparative preference statement ψ, can be used to allow pruning at a partial node in the search tree.

The algorithm for $\mathcal{Y}$-entailment is efficient for a broad range of input comparative preference statements, including those in CP-nets, CP-theories, but also allowing comparisons between partial tuples. In particular, it allows as inputs expressing a preference for one outcome over another, which is important for many applications.

In fact, the approach will be fairly efficient for many more kinds of inputs. In terms of complexity, the key issue is the efficiency of the selection-projection operation, i.e., how hard it is to test, for $\varphi \in \Gamma$, if a pair (y, y') is in φ_a^Y (for $a \in \underline{A}$ and $Y \in \mathcal{Y}$, with $Y \cap A = \emptyset$). If this is polynomial then the algorithm is polynomial (since the decomposibility property expressed by Proposition 1, means that the complexity increases only linearly with the number of input comparative preference statements $|\Gamma|$). More generally, the problem of determining if a pair (y, y') is in φ_a^Y is in NP as long as it is polynomial to test if a given pair of outcomes satisfies φ.

Consider a statement φ of the form $F > G \parallel T$, where F and G are propositional formulae (cf. [McGeachie and Doyle, 2004; Lang, 2004]). Determining if a pair (y, y') is in φ_a^Y is "only" NP-complete. Moreover, if comparative preference statement φ is elicited from a user, then one would expect that usually F and G will only involve a small number of propositional variables, in which case, we'll need to solve just a small instance of a NP-complete problem, and so may well be fairly easy.

Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under Grant No. 00/PI.1/C075, Grant No. 08/PI/I1912 and the SFI SRC project ITOBO.

References

[Boutilier *et al.*, 1999] C. Boutilier, R. Brafman, H. Hoos, and D. Poole. Reasoning with conditional *ceteris paribus* preference statements. In *Proceedings of UAI-99*, pages 71–80, 1999.

[Boutilier *et al.*, 2001] C. Boutilier, F. Bacchus, and R. Brafman. UCP-networks: A directed graphical representation of conditional utilities. In *Proceedings of UAI'01*, 2001.

[Boutilier *et al.*, 2004] 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.

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

[Brafman and Domshlak, 2002] R. Brafman and C. Domshlak. Introducing variable importance trade-offs into CP-nets. In *Proceedings of UAI-02*, pages 69–76, 2002.

[Brafman and Domshlak, 2008] R. Brafman and C. Domshlak. Graphically structured value-function compilation. *Artificial Intelligence*, 172:325349, 2008.

[Brafman *et al.*, 2006] R. Brafman, C. Domshlak, and E. Shimony. On graphical modeling of preference and importance. *Journal of Artificial Intelligence Research*, 25:389–424, 2006.

[Domshlak *et al.*, 2003] C. Domshlak, F. Rossi, K. Venable, and T. Walsh. Reasoning about soft constraints and conditional preferences: complexity results and approximation techniques. In *Proc. IJCAI03*, 2003.

[Goldsmith *et al.*, 2005] J. Goldsmith, J. Lang, M. Truszczyński, and N. Wilson. The computational complexity of dominance and consistency in CP-nets. In *Proceedings of IJCAI-05*, pages 144 –149, 2005.

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

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

[Wilson, 2004a] N. Wilson. Consistency and constrained optimisation for conditional preferences. In *Proceedings of ECAI-04*, pages 888–892, 2004.

[Wilson, 2004b] N. Wilson. Extending CP-nets with stronger conditional preference statements. In *Proceedings of AAAI-04*, pages 735–741, 2004.

[Wilson, 2006] N. Wilson. An efficient upper approximation for conditional preference. In *Proceedings of ECAI-06*, pages 472–476, 2006.