Checking Consistency of CP-Theory Preferences in Polynomial Time

Erik Rauer[1], **Samik Basu**[1], **Vasant Honavar**[2,3,4]

[1] Department of Computer Science, Iowa State University, USA
[2] Artificial Intelligence Research Laboratory
[3] Institute for Computational and Data Sciences
[4] College of Information Sciences and Technology, Pennsylvania State University, USA
{etrauer,sbasu}@iastate.edu, vuh14@psu.edu

Abstract

We investigate the problem of checking the consistency of qualitative preferences expressed in CP-theory. This problem is PSPACE-Complete even when the preferences are locally consistent or the preference variables have binary domain. We present a new sufficient condition for consistency of preferences and show that the condition can be checked in polynomial time in settings of practical relevance (locally consistent or binary domain preference variables). We further show how the resulting sufficient condition can be used to efficiently identify a subset of outcomes that are non-dominated with respect to a set of qualitative preferences.

Introduction

Representing and reasoning about preferences is central to rational decision making in a wide range of applications, from public policy to healthcare, economics, and security. Preferences can be quantitative (Keeney and Raiffa 1993; French 1986) or qualitative (Doyle and Thomason 1999). However, in many real-world applications, it is more natural to express preferences in qualitative rather than quantitative terms (Doyle and Thomason 1999). Hence, in the past two decades, there has been a growing body of work focused on languages for representing and reasoning with qualitative preferences (Brafman and Domshlak 2009; Doyle and Thomason 1999; Santhanam, Basu, and Honavar 2016; Cornelio et al. 2021).

Driving Research Question. A key challenge in reasoning with preferences has to do with the fact that stated preferences are often inconsistent, that is, they induce dominance relations over outcomes o and o' such that $o \succ o'$ and $o' \succ o$. The problem of consistency checking has been shown to be PSPACE-Complete for languages that can express conditional preferences, e.g., CP-nets (Goldsmith et al. 2008) and the more expressive CP-theories. The seminal work of Wilson (2011) investigated a logic of conditional preferences that can be used to reason about preferences expressed using CP-nets and TCP-nets and characterized several sufficient conditions (conditionally acyclic, strongly conditionally acyclic, context-uniform conditionally acyclic) for the consistency of these preferences. This work also showed that testing for these sufficient conditions is CoNP-Complete (even when the domain of preference variables is restricted to be binary). This raises an important, as yet unanswered question. Are there sufficient conditions for consistency of qualitative preferences that can be efficiently checked, e.g., in polynomial time?

Key Contributions. We answer the preceding question by identifying a new sufficient condition, *Cardinality-based Conditional Acyclicity* (`cc`-acyclic), that can be checked in polynomial time in settings of practical relevance (that is, locally consistent preferences and for preference variables with binary domain). This condition is weaker than the fully acyclic and stronger than the conditional acyclic conditions, while also being incomparable to the strong conditional and context uniform conditional acyclic conditions of Wilson (2011).

The practical importance of this result stems from its usefulness in solving problems in which the objective is to find the top k alternative in a total order that is consistent with the partial order of outcomes induced by a given set of preferences (Wang et al. 2012; Peng, Wong, and Wan 2012). The solution of such problems involves the generation of decision trees (referred to as complete search trees (Wilson 2011)) over preference variables, where leaf-level elements are organized from left to right in a total order consistent with the partial order of outcomes induced by preferences. The process used to generate such trees relies on preferences satisfying a sufficient condition for their consistency. Armed with such a sufficient condition, namely, `cc`-acyclic, that is polynomially checkable, we can use it to efficiently solve the problem of finding the top k outcomes in a total order that is consistent with the partial order of outcomes induced by a given set of CP-Theory preferences.

Wilson (2011) also showed that for preference statements satisfying the *context-uniform* sufficient condition, there exists a strict partial order extension of the dominance relation induced by the preferences; and that any dominance induced by this extension, referred to as an *upper approximation* (UA), is computable in polynomial time. We show that when preferences satisfy the `cc`-acyclic condition, we can obtain an UA in polynomial time. Such an UA offers an efficient means of identifying a subset of outcomes that are non-dominated with respect to the given preferences.

Statement ID	Preference Statement		
p_1	b_1 :	$A = a_1 \succ A = a_2$	[ ]
p_2	a_1 :	$C = c_1 \succ C = c_2$	[ ]
p_3	c_1 :	$A = a_2 \succ A = a_1$	[B]
p_4	a_1 :	$B = b_2 \succ B = b_1$	[C]
p_5	a_2 :	$C = c_2 \succ C = c_1$	[B]
p_6	a_2 :	$B = b_1 \succ B = b_2$	[ ]

Table 1: A set P of preference statements

Qualitative Preference Language

CP-nets (Boutilier et al. 2004), TCP-nets (Brafman, Domshlak, and Shimony 2006), and CP-theories (Wilson 2004b) present languages for expressing complex qualitative preferences over attributes of outcomes. The semantics of these languages use the well-studied *ceteris paribus* ("all else being equal" (Boutilier et al. 2004)) strategy to capture the dominance of an outcome over another due to a specific attribute while keeping valuations of all other (except relatively less important) attributes equal. Here, we closely follow the CP-theory-based language.

Notations. Let $\mathcal{V}$ be a finite set of variables (attributes / properties of outcomes) and let $dom(X)$ be the domain of the variable $X \in \mathcal{V}$. Given a set of variables $\mathcal{U} \subseteq \mathcal{V}$, a set of assignments to $\mathcal{U}$ is denoted by $\varrho \in \prod_{X \in \mathcal{U}} dom(X)$. We use $var(\varrho)$ to denote the set of variables assigned in ϱ. For any $X \in var(\varrho)$, we use $\varrho(X)$ to denote the valuation of X in ϱ. The set of outcomes $\mathcal{O}$ is such that any $o \in \mathcal{O}$ is a set of assignments to all variables in $\mathcal{V}$.

Definition 1 (Satisfiability & Compatibility)**.** *We say that ϱ_1 satisfies ϱ_2, denoted by $\varrho_1 \Vdash \varrho_2$, if*

$$\forall X\ [X \in var(\varrho_2)\ \Rightarrow\ (X \in var(\varrho_1)\ \wedge\ \varrho_2(X) = \varrho_1(X)]$$

We say that ϱ_1 is compatible with ϱ_2, denoted by $\varrho_1 \bowtie \varrho_2$, if

$$\forall X\ [(X \in var(\varrho_1)\ \wedge\ X \in var(\varrho_2))\ \Rightarrow\ \varrho_2(X) = \varrho_1(X)]$$

We will use upper-case letters for representing variables and the lower-case counterpart with or without subscripts to denote the valuations of the corresponding variable. For example, $A \in V$ is a variable, and a_1, a_2 are its valuations.

Qualitative Preference Statements: Syntax

A preference statement p in CP-theory (Wilson 2004b) is of the form $\varrho :\ X = x \succ X = x'\ [\Omega]$, where $X \in \mathcal{V}$, $x, x' \in dom(X)$, $\Omega \subseteq \mathcal{V}$, and sets $var(\varrho)$, $\{X\}$ and Ω are pairwise disjoint. Intuitively, a preference statement specifies that in the context of some valuations of variables ϱ (that is, conditioned on the valuation), the value of X being x is preferred to x' regardless of the valuations of the variables in Ω (variables in Ω are relatively less important than X).

For a given preference statement p in the above form, we use ϱ_p, X_p, and Ω_p to denote ϱ, X, and Ω.

Qualitative Preference Statements: Semantics

The semantics of preferences induces a dominance relationship between pairs of outcomes (Boutilier et al. 2004).

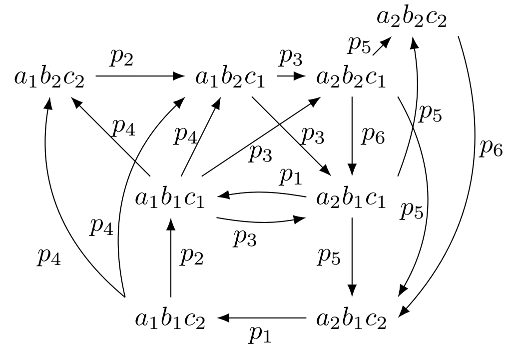


Figure 1: Induced Preference Graph over dominance relation induced by preferences in Table 1

Definition 2 (Dominance due to Preferences)**.** *Given a preference statement p of the form $\varrho :\ X = x \succ X = x'\ [\Omega]$ and two outcomes $o, o' \in \mathcal{O}$, the preferential dominance relation $o \succ o'$ is induced by p, denoted by $p \models o \succ o'$, if*

1. $o \Vdash\ \varrho\ \wedge\ o(X) = x$,
2. $o' \Vdash\ \varrho\ \wedge\ o'(X) = x'$, and
3. $\forall Y, Y \in \mathcal{V} \setminus (\{X\} \cup \Omega)\ \Rightarrow\ o(Y) = o'(Y)$.
[ceteris paribus condition]

Proceeding further, given a set of preference statements $P = \{p_1, p_2, \ldots, p_m\}$, the preferential dominance relation $o \succ o'$ is induced by P, denoted by $P \models o \succ o'$, if

1. $o = o_1 \succ o_2 \succ \ldots \succ o_k = o'$, and
2. $\forall i \in [1, k-1], \exists j \in [1, m]$ such that $p_j \models o_i \succ o_{i+1}$.

In the above, the dominance modeled by an individual preference statement is referred to as a "flip", where a flip corresponds to an improvement of valuations of a specific variable in an outcome as presented in a preference statement (Boutilier et al. 2004). The dominance modeled by a sequence of preference statements corresponds to the transitive closure of flips. This results in a directed graph where the vertices in the graph correspond to the outcomes, and the directed edge from one vertex to another captures the dominance of the destination vertex over the source vertex. This graph is referred to as the *induced preference graph* (IPG)

Example 1. *Consider the preference statements $\{p_i \mid i \in [1, 6]\}$ over $\mathcal{V} = \{A, B, C\}$ in Table 1 such that for all $X \in \mathcal{V}$, $dom(X) = \{x_1, x_2\}$. Figure 1 presents the induced preference graph for the statements. For clarity, we have not presented the transitive closure of the edge relations. We have annotated each edge with the preference statement that induces the edge. For example, the preference statement p_2 induces the edge from $a_1b_1c_2$ to $a_1b_1c_1$ and $a_1b_2c_2$ to $a_1b_2c_1$; this is because, according to p_2, when the $A = a_1$ variable C' valuation of c_1 is preferred to c_2, all else being equal (in this case the valuation of B is either b_1 for both outcomes being compared, or b_2 for both outcomes being compared).*

Consistency Testing of CP-theory Preferences

Background. A set of preference statements is said to be consistent if the dominance relation of the corresponding

IPG is a partial order. Conversely, if there is a cycle in the IPG, then the preference statements are inconsistent. A cycle in the IPG results in a scenario in which the preferences model $o \succ o'$ as well as $o' \succ o$. Because consistency testing involves testing for the absence of cycles in the IPG (where the size of the graph is exponential to the number of variables), in general, the resulting problem is PSPACE-Complete (Goldsmith et al. 2008).

There are two consistency requirements for preferences: intravariable consistency and intervariable consistency. The intravariable consistency, also referred to as *local consistency*, requires that (as per the preferences) the valuations of each variable can be arranged in a partial order under any possible valuations of the other variables. E.g., the preference statements

$$b_1 : A = a_1 \succ A = a_2\ [\,] \quad c_1 : A = a_2 \succ A = a_1\ [B]$$

are locally inconsistent because there exists a valuation of variables b_1c_1 for which $a_1 \succ a_2 \succ a_1$. Wilson (2004b) shows that the determination of local consistency is CoNP-Complete, although for preference statements P over variables $\mathcal{V}$ with small domain (e.g. binary domain), local consistency can be checked in $O(|P|^2 \cdot |\mathcal{V}|)$ time.

Intervariable consistency, on the other hand, requires that the conditional dependency and relative importance dependencies between variables should ensure a partial order over valuations of the set of variables participating in the dependencies. A straightforward albeit restrictive condition for consistency can be that preferences must be locally consistent and the transitive closure of conditional and relative importance relationship between the variables must be irreflexive. This condition is referred to as the *fully acyclic* condition (Wilson 2004b; Goldsmith et al. 2008).

Wilson (2011) introduced several sufficient conditions for the intervariable consistency of locally consistent preferences. They are *weaker* than the fully acyclic condition and the problem of checking these conditions is CoNP-complete. In contrast, we present a new sufficient condition for intervariable consistency of locally consistent preferences and prove that it can be checked in polynomial time. Our sufficient condition is stronger than the fully acyclic condition and is not comparable to the conditions proposed by Wilson.

Dependency Graph with Edge Cardinality

Our sufficient condition uses the dependency graph for CP-theory preferences. The vertices in this graph correspond to the variables over which the preferences are specified, and the directed edges capture different types of dependencies between the variables. We show that if the dependency graph satisfies a specific acyclic property, then the corresponding preference statements are guaranteed to be consistent. We proceed with the definition of a dependency graph.

Definition 3 (Dependency Graph)**.** *Given a set of preference statements P on variables $\mathcal{V}$, the dependency graph of P is a directed graph $G(P) = (\mathcal{V}, E)$ where $E = E_C \cup E_R$ is a union of the sets $E_C = \{(Y, X) \mid \exists p \in P,\ Y \in var(\varrho_p) \wedge X = X_p\}$ and $E_R = \{(X, Z) \mid \exists p \in P,\ X = X_p \wedge Z \in \Omega_p\}$. We call E_C the set of* conditional arcs *and E_R the set of* relative importance arcs.

Note that it is possible $E_C \cap E_R \neq \emptyset$. We further define a function $\mathbf{Card}_P \colon E_C \to \mathbb{N}$ that assigns a weight to each conditional edge of the dependency graph of P equal to the number of different values of the source variable that are in the condition of some preference for the target variable. Formally, given $(Y, X) \in E_C$, $\mathbf{Card}_P((Y, X)) =$

$$|\{y \in dom(Y) \mid \exists p \in P, y \in \varrho_p \wedge X = X_p\}|.$$

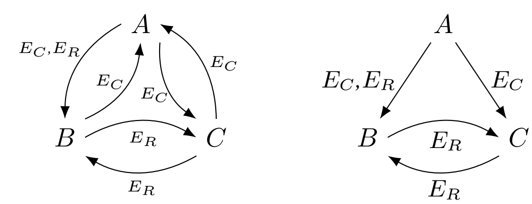
We will use the $\mathbf{Card}(.)$ when the set P for preferences is immediate in the context. Next, we describe a subgraph of $G(P)$ as follows.

Definition 4. *Given a set of preference statements P on variables $\mathcal{V}$, the high cardinality dependency graph of P is the directed graph $H(P) = (\mathcal{V}, E')$ where*

$$E' = E_R \cup \{e \in E_C \mid \mathbf{Card}_P(e) \geq 2\}$$

Note that $H(P)$ is the subgraph of $G(P)$ where all edges in $E_C \setminus E_R$ with $\mathbf{Card}_P = 1$ are removed.

Example 2. *Consider the preference statements of Table 1. The corresponding dependency graph $G(P)$ and high cardinality dependency graph $H(P)$ are:*



$G(P)$ $\qquad\qquad$ $H(P)$

The edges (B, A) and (C, A) are not in $H(P)$ since they only have $\mathbf{Card}_P$ of 1 (A is only conditionally dependent on B under b_1 due to statement p_1 and C under c_1 due to p_3).

cc-acyclic Sufficient Condition

We discuss a sufficient condition of consistency for CP-theory preferences based on the acyclicity of $H(P)$.

Theorem 1 (Cardinality-based Conditional Acyclicity)**.** *Given a set P of preference statements, if P is locally consistent and $H(P)$ is acyclic, then P is consistent. We refer to this condition as the cc-acyclic sufficient condition.*

The proof of the above theorem relies on the concept of a *complete search tree* (cs-tree) introduced in (Wilson 2006). In the following, we proceed with the definition and properties of cs-trees, followed by the proof of Theorem 1.

Definition 5 (Complete Search Tree (Wilson 2011))**.** *A complete search tree over a set of variables $\mathcal{V}$ with $dom(X)$ being the domain of $X \in \mathcal{V}$ is a tuple $(\mathcal{V}, \mathcal{N}, \mathcal{L}, var, con, val, E)$, where (a) $\mathcal{N}$ is the set of internal nodes, (b) $\mathcal{L}$ is the set of leaf nodes, (c) $var \colon \mathcal{N} \to \mathcal{V}$ is the labeling of each internal node to some variable in $\mathcal{V}$, (d) $con \colon \mathcal{N} \to \prod_{X \in \mathcal{V}' \subset \mathcal{V}} dom(X)$ is the context of each internal node described in terms of the valuations of variables in $\mathcal{V}$, (e) $val \colon \mathcal{L} \to \prod_{X \in \mathcal{V}} dom(X)$ is the valuation of all variables associated with each leaf node, and (f) $E \colon \mathcal{N} \times dom(X \mid X \in \mathcal{V}) \to \mathcal{N}$ is the edge relation that associates a node and valuation of a specific variable with another node.*

The semantics associated with a complete search tree is as follows. Each internal node is labeled with some variable. The context of an internal node ($con(n)$) is the valuations of all variables from the root of the tree to that internal node. The context for the root node is $\top$, i.e., it does not have a specific context. An internal node labeled with a variable X has $|dom(X)|$ children nodes; an edge to a child is associated with a specific valuation of X. Furthermore, for each internal node, the valuations of X (and thus the corresponding children of the node) are totally ordered from left to right. The depth of the tree is equal to $|\mathcal{V}|$. Each leaf node in the tree is associated with a unique assignment to all variables in $\mathcal{V}$, which corresponds to the assignments on the path from the root to that leaf node. These valuations are totally ordered from left to right at the leaf level. For any leaf l in a cs tree, $val(l)$ can be interpreted as an outcome. Thus, a cs tree over $\mathcal{V}$ induces a total order on the outcomes defined over the variables $\mathcal{V}$. Given a set P of preference statements over $\mathcal{V}$, a complete search tree cs, described over $\mathcal{V}$ satisfies P if the total order at the leaf level of cs extends the partial order of dominance relations induced by P. This condition is equivalent to that of Wilson (2011) :

Proposition 1 (Complete Search Tree Satisfying Preference Statements (Wilson 2011))**.** *A complete search tree cs satisfies a set of preference statements P if and only if the following two conditions hold for all $p \in P$ of the form $\varrho_p : X_p = x_p \succ X_p = x'_p[\Omega_p]$:*

1. *for any outcome o such that $o \Vdash \varrho_p$, on the path from the root of cs to leaf l with $val(l) = o$, X_p appears before each element of Ω_p,*
2. *for any internal node n, if $X_p = var(n)$ and $\varrho_p \bowtie con(n)$ (i.e., ϱ_p and $con(n)$ are compatible, see Definition 1), then x_p appears to the left of x'_p in the total ordering of outgoing children of n.*

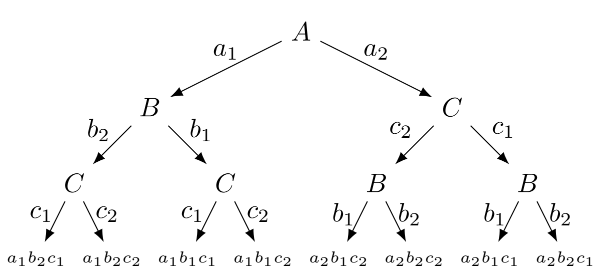
Finally, Wilson presents the result that characterizes consistency in terms of cs-tree, called the (weak) conditional acyclic condition:

Lemma 1 (Wilson 2011)**.** *If there exists a cs-tree that satisfies preference statements P then P is consistent.*

Example 3. *Consider the locally consistent preference statements p_4 and p_5 from Table 1:*

$$a_1 : B = b_2 \succ B = b_1\ [C] \quad a_2 : C = c_2 \succ C = c_1\ [B]$$

A cs-tree that satisfies the above preference statements is



The variable A at the root node does not have a context as per the preference statements. In the left sub-tree, the context for the node with variable B is the valuation a_1 of A. Hence, in this context, the node labeled with variable C appears after the node labeled with B (C being relatively less important as per the first preference statement p_4). Additionally, the ordering of the children of this node is $b_2 > b_1$ which follows preference statement p_4.

We can now present the proof for Theorem 1. The central idea is to show that for locally consistent preferences P with acyclic $H(P)$, one can construct a cs-tree that satisfies P (as per Proposition 1) and therefore, by Lemma 1, P is consistent.

Proof of Theorem 1. $H(P)$ is acyclic according to the $\mathtt{cc}$ -acyclic condition, so there exists a total order on the variables $\mathcal{V}$ based on $H(P)$. Let $Y_1, \ldots, Y_{|\mathcal{V}|}$ be this total order. We construct the complete search tree as follows. For each internal node n at level i, $var(n) = Y_i$. For any such internal node n at level i and variable Y_j with $e = (Y_j, Y_i)$ as an edge in $G(P)$ and $\mathbf{Card}_P(e) = 1$, there exists a single valuation $a \in dom(Y_j)$ such that any preference statement $p \in P$ with $X_p = Y_i$ and $Y_j \in var(\varrho_p)$ has $\varrho_p(Y_j) = a$. We denote all such valuations of Y_j's for preference statements over Y_i by α_i. For all other variables Y_j with $j < i$ that are not in $var(\alpha_i)$, we denote their valuations in $con(n)$ by $\beta(n)$.

$$\text{Finally,} \quad \gamma(n) = \alpha_i \cup \beta(n) \tag{1}$$

Consider the relation $>_\omega^X \subseteq dom(X) \times dom(X)$ where ω is the assignment to variables in $W \subseteq \mathcal{V} \setminus X$. Let $>_\omega^X$ be defined as the transitive closure of all pairs (x, x') for which there exists a preference statement of the form $\varrho : X = x \succ X = x'\ [\Omega]$ such that $\omega \Vdash \varrho$. For locally consistent preferences, for any X, it is immediate that the relation $>_\omega^X$ is irreflexive. We can therefore arrange the children of n (total) ordered according to the relation $>_{\gamma(n)}^{Y_i}$.

We now show that the constructed cs-tree satisfies conditions (1.) and (2.) of Proposition 1. Consider a preference statement $p \in P$ of the form $\varrho_p : X_p = x_p \succ X_p = x'_p\ [\Omega_p]$. Let the variable X_p be the i-th variable in the total order used to construct the cs-tree, i.e., X_p is Y_i. And Y_i is the variable at the node n of the cs-tree.

For any variable $Z \in \Omega_p$, if Z is Y_j, $(Y_i, Y_j) \in E_R$, and therefore $i < j$ in our total order. In other words, the assignment to Y_i is always decided before the assignment to Y_j in the cs-tree, thus satisfying condition (1.) of Proposition 1.

Let node n be such that $var(n) = X_p$ and $\varrho_p \bowtie con(n)$ (compatible). Consider $Y_j \in \mathcal{V}$ whose valuation a is present in ϱ_p. Then $e = (Y_j, Y_i) \in E_C$ and there are two possibilities for the cardinality of e: $\mathbf{Card}_P(e) = 1$ or $\mathbf{Card}_P(e) \geq 2$.

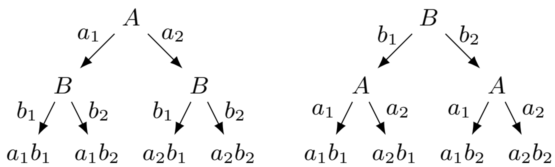
- If $\mathbf{Card}_P(e) = 1$, then $a \in \alpha_i \subseteq \gamma(n)$ by definition.
- On the other hand, if $\mathbf{Card}_P(e) \geq 2$, then $e \in H(P)$ so $j < i$ by choice of the total order. Since $\varrho_p \bowtie con(n)$, $a \in con(n)$ and $a \in \varrho_p$ as well. So $a \in \beta(n) \subseteq \gamma(n)$ by definition of $\beta(n)$.

In either case ($\mathbf{Card}_P(e) = 1$ and $\mathbf{Card}_P(e) \geq 2$) $a \in \gamma(n)$ making $\varrho_p \bowtie \gamma(n)$. Therefore, the order of Y_i according to the relation $>_{\gamma(n)}^{Y_i}$ satisfies the ordering where x_p appears to the left of x'_p in the total ordering of valuations of Y_i for the node n. Thus, condition (2.) of Proposition 1 is satisfied. $\square$

Example 4. *Consider the preference statements p_1 and p_6 from Table 1:*

$$b_1 : \; A = a_1 \succ A = a_2 \, [\,] \quad a_2 : \; B = b_1 \succ B = b_2 \, [\,] \quad (2)$$

*The edge relation that captures the (mutual) conditional dependency between the variables in these preference statements is such that **Card**$((A, B)) = 1$ and **Card**$((B, A)) = 1$. Hence, the high cardinality dependency graph (see Definition 4) has no edges (acyclic)–thus satisfying the cc-acyclic condition. The cs-trees for the above preference statements:*



Checking cc-acyclic Condition

We present here a poly-time algorithm to address the problem of determining whether the cc-acyclic condition holds.

Proposition 2. *Given a set of locally consistent preferences P, checking for the cc-acyclic sufficient conditions can be done in polynomial time.*

Proof. Observe that the dependency graph $G(P)$ contains a set of edges E where $|E| \in O(|\mathcal{V}| \cdot |P|)$; hence, the cardinality of the edges can be set in $O(|\mathcal{V}| \cdot |P|)$ time. Upon building the dependency graph $G(P)$, $H(P)$ can easily be obtained in $O(|\mathcal{V}| \cdot |P|)$ time by removing all edges in $E_C \setminus E_R$ with $\textbf{Card}_P = 1$. From there, one must simply check whether $H(P)$ is acyclic, which can be done by DFS in $O(|E| + |\mathcal{V}|) = O(|\mathcal{V}| \cdot |P| + |\mathcal{V}|)$ time. $\square$

Finding the Top *k* Outcomes

In some applications, it may be necessary to identify the top k elements in a total ordering of the outcomes that is consistent with the preferences (where k is considerably smaller than the total number of outcomes) (Peng, Wong, and Wan 2012; Wang et al. 2012).

The cs-tree-based organization of outcomes plays a role in realizing effective and efficient algorithms for these problems (Brafman, Domshlak, and Shimony 2006; Wilson 2006). For finding the top k elements, one can construct the cs-tree in a depth-first fashion (aligning with the preorder traversal of the tree) and as soon as k leaf nodes are generated, terminate the algorithm and return the outcomes at the leaf nodes. Armed with an efficient algorithm for testing whether cc-acyclic condition holds, if cc-acyclic is found to hold, we can generate a cs-tree with an appropriate number of leaf-level nodes to identify the top k outcomes.

Polynomial Upper Approximation

For preferences that are consistent and hence have a satisfying cs-tree, Wilson (2006) also introduced the notion of upper approximation, a partial-order extension of the dominance relations induced by the preferences. We next show that for preferences that satisfy the cc-acyclic condition, we can realize a polynomial-time computable upper approximation.

Definition 6 (Polynomial Upper Approximation (Wilson 2006))**.** *A binary relation $\gg$ on outcomes is an upper approximation for a set of preference statements P if for any pair of outcomes o, o': $P \vDash o \succ o' \;\Rightarrow\; o \gg o'$.*

If determining whether $o \gg o'$ can be done in polynomial time for any pair of outcomes, $\gg$ is said to be a polynomial upper approximation.

The upper approximation of P defined in terms of cs-tree is as follows.

Proposition 3 (Wilson 2011)**.** *Let $\mathcal{R}$ be some subset of complete search trees that satisfy a set of preference statements P. The relation $\gg^{\mathcal{R}}$ is an upper approximation of P, where, for all outcome-pairs o, o', $o \gg^{\mathcal{R}} o'$ if and only if for every cs-tree $T \in \mathcal{R}$, o is preferred to o' in T's total order.*

We now proceed to introduce some necessary concepts and notations before presenting the polynomial-time upper approximation that can be computed for preferences satisfying the cc-acyclic condition.

Definition 7. *Let P be a set of preference statements. Consider outcomes o, o' such that $o \neq o'$.*

- *$\Delta(o, o')$ is the set of variables where o and o' differ, that is, $\Delta(o, o') = \{X \in \mathcal{V} \mid o(X) \neq o'(X)\}$.*
- *$\Theta(o, o')$ is the set of variables in $\Delta(o, o')$ where all of their parents in the transitive closure of edges in $H(P)$ are not in $\Delta(o, o')$, i.e. if $H^*(P)$ is the transitive closure of edges in $H(P)$, then*
$$\Theta(o, o') = \{X \in \Delta(o, o') \mid \forall (Y, X) \in H^*(P), Y \notin \Delta(o, o')\}$$
- *For $X \in \Theta(o, o')$, we use $PA(o, o', X)$ to denote the set consisting of the parents of X in $H^*(P)$, i.e.,*
$$PA(o, o', X) = \{Y \in \mathcal{V} \mid (Y, X) \in H^*(P)\}$$
We use $\mathtt{pa}(o, o', X)$ to denote the valuations of variables in $PA(o, o', X)$ in o,
$$\mathtt{pa}(o, o', X) = \{y \in dom(Y) \mid Y \in PA(o, o', X) \;\wedge\; o(Y) = y\}$$
- *For each parent Y of X in $G(P)$ with **Card**$_P = 1$, $\mathtt{c}(X)$ contains the valuation of Y present in the condition of the preference statements $p \in P$ of the form*
$$\varrho_p : \; X = x_p \succ X = x'_p \, [\Omega_p] \text{ That is,}$$
$$\mathtt{c}(X) = \left\{ y \in dom(Y) \left| \begin{array}{l} (Y, X) \in E_C \setminus E_R \\ \textbf{Card}_P((Y, X)) = 1 \\ \exists p \in P, X_p = X \;\wedge\; \varrho_p(Y) = y \end{array} \right. \right\}$$
- *Finally, define $\mathtt{d}(X)_{o,o'} = \mathtt{pa}(o, o', X) \cup \mathtt{c}(X)$.*

The following lemma follows from the definitions of $\mathtt{d}(.)$ (Defintion 7) and $\gamma(.)$ (Equation 1 in Proof of Theorem 1).

Lemma 2. *Given a set P of locally consistent preference statements over variables $\mathcal{V}$ that satisfy the cc-acyclic condition, consider a cs-tree with the variable ordering $Y_1, Y_2, \ldots, Y_{|\mathcal{V}|}$ constructed following the steps described in the proof of Theorem 1. For any outcomes o and o' such that the first variable on which they differ is Y_i, then for node n in the cs-tree, where $var(n) = Y_i$ and $o \Vdash con(n)$, the relation $>^{Y_i}_{\gamma(n)}$ is equivalent $>^{Y_i}_{\mathtt{d}(Y_i)_{o,o'}}$.*

Proof. Let $p \in P$ be a preference statement of the form $\varrho_p : X_p = x_p \succ X_p = x'_p\ [\Omega_p]$, such that X_p is Y_i and $\gamma(n) \Vdash \varrho_p$. Since Y_i is the first variable in the ordering where o, o' differ, $Y_i \in \Theta(o, o')$. Consider $Y_j \in var(\varrho_p)$ with valuation of Y_j in ϱ_p being y. Hence, $y \in \gamma(n)$ (see Equation 1). If $y \in \alpha_i$, then $\mathbf{Card}((Y_j, Y_i)) = 1$ and $y \in \mathtt{c}(Y_i)$ (see Definition 7) making $y \in \mathtt{d}(Y_i)_{o,o'}$. Furthermore, if $y \in \beta(n)$, then the $\mathbf{Card}((Y_j, Y_i)) \geq 2$. Hence, by definition of $PA(o, o', Y_i)$ (Def. 7), $Y_j \in PA(o, o', Y_i)$ and $y \in \mathtt{pa}(o, o', Y_i)$. This implies, $y \in \mathtt{d}(Y_i)_{o,o'}$.

A similar argument shows that the same holds for $\gamma(n)$ and $p \in P$ with $\mathtt{d}(Y_i)_{o,o'} \Vdash \varrho_p$. Hence, the relations $>^{Y_i}_{\mathtt{d}(Y_i)_{o,o'}}$ and $>^{Y_i}_{\gamma(n)}$ are equivalent. □

Definition 8 (Upper Approximation)**.** *Let P be a set of preference statements that satisfy the* cc *-acyclic sufficient conditions. Note that this means that $H(P)$ is acyclic and therefore $\Theta(o, o')$ must be non-empty if $o \neq o'$. Define the binary relation $\gg^{cc}$ by: $o \gg^{cc} o'$ if and only if for all $X \in \Theta(o, o')$, $o(X) >^X_{\mathtt{d}(X)_{o,o'}} o'(X)$.*

As mentioned, this is an upper approximation of P.

Theorem 2. *Let P be a set of preference statements that satisfy the* cc *-acyclic sufficient condition. The relation $\gg^{cc}$ defined above is an upper approximation of P.*

Proof. By the cc-acyclic conditions $H(P)$ is acyclic, so there exists some nonempty set of total orders on the variables of $\mathcal{V}$, each of which satisfies the partial order over vertices induced by $H(P)$ (i.e., the total order is some topological ordering of the vertices in acyclic graph $H(P)$). For each such a total order, one can construct a cs-tree in a manner similar to that described in the proof of Theorem 1.

Let $\mathcal{R}_{cc}$ be the set of all such cs-trees that can be constructed as described in the Theorem 1. We will show that $\gg^{cc}$ is equivalent to the relation $\gg^{\mathcal{R}_{cc}}$ and is thus an upper approximation by Proposition 3.

Case: $\gg^{cc} \Rightarrow \gg^{\mathcal{R}_{cc}}$ Consider an arbitrary *cs*-tree $T \in \mathcal{R}_{cc}$ and let o, o' be a pair of outcomes such that $o \gg^{cc} o'$. Consider the total order $Y_1, \ldots, Y_{|\mathcal{V}|}$ on the variables used to construct T. Let Y_i be the first variable in this order where o and o' differ, i.e. for all $j < i$, $o(Y_j) = o'(Y_j)$. Since o, o' differ on Y_i, $Y_i \in \Delta(o, o')$ and for all $j < i$, $Y_j \notin \Delta(o, o')$. Thus by the choice of total order, $Y_i \in \Theta(o, o')$ (all parents of Y_i in the transitive closure of $H(P)$ must be a Y_j with $j < i$). Then $o(Y_i) >^{Y_i}_{\mathtt{d}(Y_i)_{o,o'}} o'(Y_i)$ by the definition of $\gg^{cc}$. By Lemma 2, $>^{Y_i}_{\mathtt{d}(Y_i)_{o,o'}}$ and $>^{Y_i}_{\gamma(n)}$ are equivalent, therefore, $o(Y_i) >^{Y_i}_{\gamma(n)} o'(Y_i)$. Since Y_i is the highest variable in T where o and o' differ, o is more preferred to o' according to the total order of the leaves of T. **Case:** $\gg^{\mathcal{R}_{cc}} \Rightarrow \gg^{cc}$ Let o, o' be a pair of outcomes such that $o \gg^{cc} o'$ does not hold. By definition of $\gg^{cc}$, there exists some $Y \in \Theta(o, o')$ such that $o(Y) >^Y_{\mathtt{d}(Y)_{o,o'}} o'(Y)$ does not hold. Recall that the *cs*-trees of $\mathcal{R}_{cc}$ are constructed with some variable ordering $(Y_1, \ldots, Y_{|\mathcal{V}|})$ that conforms to the high cardinality graph $(H(P))$. Note that there are many such ordering; consider the ordering where $Y (= Y_i)$ appears the earliest (i.e., i is minimal) and construct a *cs*-tree T with that ordering. Let n be the node in T with $var(n) = Y_i$ and $o \Vdash con(n)$. Since $Y_i \in \Theta(o, o')$, any Y_j with $j < i$ has $Y_j \notin \Delta(o, o')$. Proceeding further, based on the Lemma 2, $o(Y) >^Y_{\mathtt{d}(Y)_{o,o'}} o'(Y)$ not holding implies $o(Y) >^Y_{\gamma(n)} o'(Y)$ does not hold. Since n was the first internal node where o, o' differ, o does not appear to the left of o' at the leaf-level of T. □

Example 5. *In Example 4, the locally consistent preference statements in Equation 2 induce the following dominance $a_1b_1 \succ a_2b_1$, $a_2b_1 \succ a_2b_2$ and $a_1b_1 \succ a_2b_2$. We have also established that the preference statements satisfy the* cc*-acyclic condition and have two cs-trees (satisfying the preference statements) that are constructed using the strategy described in Theorem 1. The upper approximation of the preference statements is defined using the relation $\gg^{cc}$.*

Consider, for example, the outcomes $o = a_1b_1$ and $o' = a_2b_2$. The set of variables over which the outcomes are different $\Delta(o, o') = \{A, B\}$. Recall that the high dependency graph $H(.)$ for these preferences is such that the variables A and B do not have parents. We conclude that $\Theta(o, o') = \{A, B\}$. Proceeding further, $\mathtt{pa}(o, o', A) = \emptyset$, $\mathtt{c}(A) = \{b_1\}$ and $\mathtt{d}(A)_{o,o'} = \{b_1\}$; while $\mathtt{pa}(o, o', B) = \emptyset$, $\mathtt{c}(B) = \{a_1\}$ and $\mathtt{d}(B)_{o,o'} = \{a_2\}$. Therefore, $o(A) >_{\mathtt{d}(A)_{o,o'}} o'(A)$ according to the first preference statement (Equation 2) and $o(B) >_{\mathtt{d}(B)_{o,o'}} o'(B)$ according to the second preference statement (Equation 2). Hence, $o \gg^{cc} o'$.

Using the same arguments, we will have

$$a_1b_1 \gg^{cc} a_2b_2,\ \ a_2b_1 \gg^{cc} a_2b_2,\ \ a_1b_2 \gg^{cc} a_2b_2,$$
$$a_1b_1 \gg^{cc} a_2b_1,\ \ a_1b_1 \gg^{cc} a_1b_2$$

The above partial order is the intersection of the total orders captured by all the cs-trees presented in Example 4.

Proposition 4. *Given a set P of locally consistent preferences that satisfy* cc*-acyclic condition, $\gg^{cc}$ relation can be computed in polynomial time.*

Proof. The runtime complexity for $\gg^{cc}$ depends on the computation of the relations described in Definition 7.

$\Delta(o, o')$ can be computed in $O(|\mathcal{V}|)$ time. As $G(P)$ and thus $H(P)$ can be constructed in $O(|\mathcal{V}|\cdot|P|)$ time, $\Theta(o, o')$ requires exploration of $H(P)$ for each $X \in \Delta(o, o')$ and thus takes $O(|\mathcal{V}|\cdot|P|)$ time for each variable. Similarly, for any $X \in \Theta(o, o')$, $PA(o, o', X)$ can also be computed in $O(|\mathcal{V}|\cdot|P|)$ time. Proceeding further, $\mathtt{pa}(o, o', X)$ can be found in $O(|\mathcal{V}|)$. Additionally, $\mathtt{c}(X)$ also requires a traversal of $G(P)$ and so can be done in $O(|\mathcal{V}|\cdot|P|)$ time. Hence, $\mathtt{d}(X)_{o,o'}$ requires $O(|\mathcal{V}|\cdot|P|)$ time. Since $\mathtt{d}(X)_{o,o'}$ must be calculated for the $|\Theta(o, o')| \in O(|\mathcal{V}|)$ variables, the run time to calculate $\mathtt{d}(.)_{o,o'}$ for all variables is $O(|\mathcal{V}|^2\cdot|P|)$.

Finally, determining whether $o(X) >^X_{\mathtt{d}(X)_{o,o'}} o'(X)$ requires checking all preference statements $p \in P$ with $X_p = X$. This results in a total runtime of $O(|\mathcal{V}|^2\cdot|P|^2)$. □

Finding a Non-Dominated Set

In some applications, preferences are used to identify a set of (non-dominated) outcomes that are not less preferred to any other outcome. Formally, $\{o \in \mathcal{O} \mid\ \nexists\, o',\ P \models o' \succ o\}$.

To find the non-dominated set of outcomes, one can use a *cs*-tree and generate the leaf-level outcomes from left to right. The left-most leaf-level outcome is guaranteed to be a non-dominated element and will be added to the solution set. For any new leaf-level element o, we then need to check whether o is dominated by any element already present in the solution set. If the answer is affirmative, then o does not belong to the solution; otherwise, it is added to the solution set. In short, the non-dominated set can be computed by iteratively adding outcomes to a partial solution set (initialized to an empty set) and each addition requires resolving dominance queries. Dominance testing for CP-theory based preferences is PSPACE-Complete (Goldsmith et al. 2005) and hence, identifying the non-dominated outcomes is a computationally expensive process. Observe that, the upper approximation relation $\gg$ (Definition 6) is such that

$$\{o \in \mathcal{O} \mid \nexists\, o',\ o' \gg o\} \subseteq \{o \in \mathcal{O} \mid \nexists\, o',\ P \models o' \succ o\}.$$

Therefore, one can use the $\gg$ relation instead of the $\succ$ relation to identify some subset of a nondominated solution. We have presented $\gg^{\mathtt{cc}}$, a polynomial time computable upper approximation for preferences satisfying cc-acyclic sufficient condition, and hence can be used to efficiently compute a subset of the non-dominated set.

Comparison with Other Sufficient Conditions

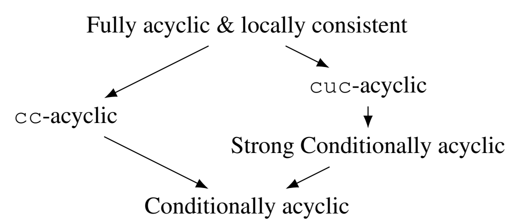
Wilson et al. (2004b) introduced the fully acyclic sufficient condition for consistency (locally consistent and $G(P)$ acyclic), which ensures the existence of partial conditional lexicographic (pcl) ordering over variables. Such a pcl ordering implies the consistency of the stated preferences. Since $H(P)$ is a subgraph of $G(P)$, an acyclic $G(P)$ implies an acyclic $H(P)$. So, the fully acyclic sufficient condition implies the cc-acyclic sufficient condition (but not vice versa).

Subsequent work by Wilson (2004a, 2011) formalized different types of sufficient conditions for CP-theory preferences: weak, strong, and context-uniform conditional acyclic conditions, and characterizes them in terms of complete search trees. The *weak conditional acyclic* condition is satisfied by preferences if there exists some *cs*-tree that satisfies the preferences. It follows from the proof of Theorem 1 that the cc-acyclic condition implies the existence of a satisfiable *cs*-tree (and ensures the weak conditional acyclicity).

The *strong conditional acyclic* condition requires that for any assignment to some subset of variables, there exists an unassigned variable X that is not conditionally dependent on and is not relatively less important than the unassigned variables. This implies the existence of a variable that is not dependent on assignments to any other variables and such a variable can be used as a root of a *cs*-tree. Similarly, the root of a subtree can be identified by considering the assignments to the variables appearing above the subtree in the *cs*-tree. In short, the strong conditional acyclic condition implies the weak acyclic condition. Wilson also introduced an even stronger sufficient condition: the *context uniform conditionally acyclic* (*cuc*) condition, which requires conditional and relative importance acyclicity in the context of each outcome. This is similar to the conditional acyclicity of TCP nets (Brafman, Domshlak, and Shimony 2006). They showed that the TCP-net preferences are consistent if all possible dependency graphs on the variables corresponding to different valuations of variables that determine the "direction" of conditional relative importance are acyclic.

We show that the cc-acyclic condition neither implies nor is implied by either the strong acyclic condition or the cuc-acyclic condition: The preference statements in Equation 2 satisfy the cc-acyclic condition as shown in Example 4. However, no cs-tree strongly satisfies these statements since both A and B are conditionally dependent on the other, so no root can be chosen for a *cs*-tree that strongly satisfies the statements. As the cuc-acyclic condition is stronger than the strong conditional acyclic condition, the preference statements do not satisfy the cuc -acylic condition either. Therefore, the cc-acyclic condition implies neither the strong conditional acyclic condition nor the cuc-acyclic condition. To see the converse, recall Example 3: for any outcome o, let the valuation of A be a_1 or a_2. When $o(A) = a_1$, the dependency relationship between the variables is $\{(A, B), (A, C), (B, C)\}$, when $o(A) = a_2$, the dependency relationship is $\{(A, B), (A, C), (C, B)\}$. In either case, the dependencies are acyclic. So, these preference statements satisfy the cuc-acyclic condition and by definition also satisfy the strong acyclic condition. However, the high-cardinality graph H for the above preferences includes the edges $\{(B, C), (C, B)\}$ that result in a cycle. This does not satisfy the cc-acyclic condition. Hence, the cuc-acyclic condition does not imply the cc-acyclic condition.

Based on the above observations, we have the following implication order among the sufficient conditions:



However, checking both the strong acyclic and cuc-acyclic conditions are CoNP-Complete for locally consistent preferences (and for binary domain variables), while the cc-acyclic condition can be checked in polynomial time.

Summary and Future Directions

We investigated the problem of checking the consistency of qualitative preferences expressed in languages based on CP-theory. We identified cc-acyclic, a new sufficient condition for the consistency of preferences, and show that the condition can be checked in polynomial time in settings of practical relevance (locally consistent or binary domain preference variables). We showed how cc-acyclic can be used to efficiently identify a subset of outcomes that are non-dominated with respect to a set of preferences. We plan to investigate whether the sufficient condition can be relaxed further without sacrificing the poly-time testability. We also plan to implement and empirically compare the cc-acyclic condition and the resulting upper approximation with those proposed by (Wilson 2011).

Acknowledgements

This work was supported in part by NSF grants IIS 2225823 and IIS 2225824.

References

Boutilier, C.; Brafman, R. I.; Domshlak, C.; Hoos, H. H.; and Poole, D. 2004. CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. *Journal of Artificial Intelligence Research*, 21: 135–191.

Brafman, R.; and Domshlak, C. 2009. Preference Handling - An Introductory Tutorial. *AI magazine*, 30(1).

Brafman, R. I.; Domshlak, C.; and Shimony, S. E. 2006. On Graphical Modeling of Preference and Importance. *Journal of Artificial Intelligence Research*, 25: 389–424.

Cornelio, C.; Goldsmith, J.; Grandi, U.; Mattei, N.; Rossi, F.; and Venable, K. B. 2021. Reasoning with PCP-Nets. *Journal of Artificial Intelligence Research*, 72: 1103–1161.

Doyle, J.; and Thomason, R. H. 1999. Background to Qualitative Decision Theory. *AI magazine*, 20: 55–68.

French, S. 1986. *Decision Theory: An Introduction to the Mathematics of Rationality*. Ellis Horwood Limited.

Goldsmith, J.; Lang, J.; Truszczynski, M.; and Wilson, N. 2005. The computational complexity of dominance and consistency in CP-nets. In *Proceedings of International Joint Conference on Artificial Intelligence*, 144–149.

Goldsmith, J.; Lang, J.; Truszczynski, M.; and Wilson, N. 2008. The Computational Complexity of Dominance and Consistency in CP-Nets. *JAIR*, 33: 403–432.

Keeney, R. L.; and Raiffa, H. 1993. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Cambridge University Press.

Peng, Y.; Wong, R. C.-W.; and Wan, Q. 2012. Finding Top-k Preferable Products. *IEEE Transactions on Knowledge and Data Engineering*, 24(10): 1774–1788.

Santhanam, G. R.; Basu, S.; and Honavar, V. 2016. Representing and Reasoning with Qualitative Preferences: Tools and Applications. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(1): 1–154.

Wang, H.; Zhou, X.; Chen, W.; and Ma, P. 2012. Top-k retrieval using conditional preference networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, 2075–2079. New York, NY, USA: Association for Computing Machinery. ISBN 9781450311564.

Wilson, N. 2004a. Consistency and constrained optimisation for conditional preferences. In *Proceedings of the 16th European Conference on Artificial Intelligence*, ECAI'04, 888–892. NLD: IOS Press. ISBN 9781586034528.

Wilson, N. 2004b. Extending CP-nets with stronger conditional preference statements. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, 735–741. AAAI Press. ISBN 0262511835.

Wilson, N. 2006. An Efficient Upper Approximation for Conditional Preference. In *Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy*, volume 141, 472–476.

Wilson, N. 2011. Computational techniques for a simple theory of conditional preferences. *Artificial Intelligence*, 175(7): 1053–1091. Representing, Processing, and Learning Preferences: Theoretical and Practical Challenges.