Towards Fast Algorithms for the Preference Consistency Problem Based on Hierarchical Models

Anne-Marie George, Nic Wilson, Barry O'Sullivan
Insight Centre for Data Analytics, School of Computer Science and IT
University College Cork, Ireland
{annemarie.george, nic.wilson, barry.osullivan}@insight-centre.org

Abstract

In this paper, we construct and compare algorithmic approaches to solve the Preference Consistency Problem for preference statements based on hierarchical models. Instances of this problem contain a set of preference statements that are direct comparisons (strict and non-strict) between some alternatives, and a set of evaluation functions by which all alternatives can be rated. An instance is consistent based on hierarchical preference models, if there exists an hierarchical model on the evaluation functions that induces an order relation on the alternatives by which all relations given by the preference statements are satisfied. Deciding if an instance is consistent is known to be NP-complete for hierarchical models.

We develop three approaches to solve this decision problem. The first involves a Mixed Integer Linear Programming (MILP) formulation, the other two are recursive algorithms that are based on properties of the problem by which the search space can be pruned. Our experiments on synthetic data show that the recursive algorithms are faster than solving the MILP formulation and that the ratio between the running times increases extremely quickly.

1 Introduction

In many fields like recommender systems and multi-objective optimization, one wants to reason over user preferences. In these problems, it is often difficult or excessively time-consuming to elicit all user preferences. The Preference Deduction Problem (PDP) aims at eliciting only few preferences and inferring more preferences from the given ones; this might then be used in a conversational recommender system, for example, to help choose which items to show to the user next [Bridge and Ricci, 2007; Trabelsi *et al.*, 2011]. Typically, an assumption is made on the type of order relation that the user (implicitly) uses to express the preference statements. Such order relations can be, e.g., comparing alternatives by the values of the evaluation functions lexicographically [Wilson, 2014], by Pareto order, weighted sums [Figueira *et al.*, 2005], based on hierarchical models [Wilson *et al.*, 2015] or by conditional preferences structures as CP-nets [Boutilier *et al.*, 2004] and partial lexicographic preference trees [Liu and Truszczynski, 2015]. Here, the choice of the order relation can lead to stronger or weaker inferences and can make solving PDP computationally more or less challenging. In a recommender system or in a multi-objective decision making scenario, the user should only be presented with a relatively small number of solutions, hence, a strong order relation is required. Using PDP based on a lexicographic models has been shown to be successful in reducing the number of solutions, however, computationally can be more expensive. See [Marinescu *et al.*, 2013] and [George *et al.*, 2015] for comparisons between order relations in a multi-objective optimization framework. While PDP based on hierarchical models yields an even lower number of solutions, it is coNP-complete and computationally expensive. The approach taken by the Preference Deduction Problem contrasts with learning a single model that coincides with the user preferences as in [Fürnkranz and Hüllermeier, 2010; Dombi *et al.*, 2007; Flach and Matsubara, 2007; Bräuning and Hüllermeier, 2012; Booth *et al.*, 2010].

In this paper, we concentrate on another problem that helps solving PDP: the Preference Consistency Problem (PCP). This is the problem of deciding whether given user preferences are consistent, i.e., not contradicting each other. In terms of hierarchical preference models, PCP determines whether there exists a hierarchical model of evaluation functions such that the induced order relation on the alternatives satisfies all preference statements given by the user. For hierarchical preference models, PCP is NP-complete and PCP and PDP are mutually expressible [Wilson *et al.*, 2015], i.e., PCP can be solved directly by solving PDP and vice versa. The main issue in this paper is to find fast algorithms to solve PCP (and thus PDP) for hierarchical models by exploiting the problem's structure and to compare their running times on a set of synthetic data.

The next section gives an introduction to hierarchical preference models, their induced order relation on alternatives, and the Preference Consistency Problem. Section 3 gives a Mixed Integer Linear Programming formulation for PCP that will function as a baseline for our runtime comparisons. Section 4 discusses the exploitation of properties of PCP instances yielding two recursive algorithms. In Section 5, the conducted experiments and their outcome are described. The last section concludes.

A longer version of this paper including proofs and examples is available online [George *et al.*, 2016].

2 The Consistency Problem Based on Hierarchical Models

For a set of preference statements, consistency is defined to be the property that the statements do not contradict each other. To formally define this term, we first describe the concept of hierarchical models.

2.1 Hierarchical Models

Hierarchical models will from here on be called HCLP models, where HCLP stands for "Hierarchical Constraint Linear Program" and points out the resemblance of the hierarchical order of the evaluation functions in HCLP models to the order of soft constraints in an HCLP [Wilson and Borning, 1993]. In the following, we define HCLP structures, HCLP models, and their implied order relation that is a kind of lexicographic order.

Definition 2.1 (HCLP structures)**.** *An HCLP structure is a triple $\langle \mathcal{A}, \oplus, \mathcal{C} \rangle$. Here, $\mathcal{A}$ is a finite set of alternatives and $\mathcal{C}$ is a set of evaluation functions from the alternatives $\mathcal{A}$ to the non-negative rational numbers $\mathbb{Q}^{\geq 0}$. $\oplus$ is an associative, commutative, and strictly monotonic operation on $\mathbb{Q}^{\geq 0}$, i.e., $x \oplus y < z \oplus y$ if and only if $x < z$.*

In an HCLP structure, the evaluations $\mathcal{C}$ as well as their combinations by $\oplus$ provide ratings of the alternatives due to unfavorable aspects, e.g., costs, for which we assume that 0 is the best value. The notion of HCLP structures was first introduced in [Wilson *et al.*, 2015] with $\oplus$ as an associative, commutative and monotonic operation. In this paper, we consider $\oplus$ to be a *strictly* monotonic operation as this yields interesting properties allowing us to formulate fast algorithms for checking consistency. Furthermore, we assume the operation to be computable in logarithmic time. These assumptions still allow interesting operators like addition and multiplication which seem natural for combining aspects like costs and distances, but strict monotonicity excludes a minimum or maximum operator which could be desired sometimes.

For a subset $C \subseteq \mathcal{C}$ of evaluations, we define the weak order (transitive and complete binary order) $\preccurlyeq_C^\oplus$ on the set of alternatives in the following way: for $\alpha, \beta \in \mathcal{A}$, $\alpha \preccurlyeq_C^\oplus \beta$ if and only if $\bigoplus_{c \in C} c(\alpha) \leq \bigoplus_{c \in C} c(\beta)$. The corresponding strict order $\prec_C^\oplus$ is given by $\alpha \prec_C^\oplus \beta$ if and only if $\alpha \preccurlyeq_C^\oplus \beta$ and $\alpha \not\succcurlyeq_C^\oplus \beta$, i.e., $\bigoplus_{c \in C} c(\alpha) < \bigoplus_{c \in C} c(\beta)$. Then, the equivalence relation $\equiv_C^\oplus$ is given by $\alpha \equiv_C^\oplus \beta$ if and only if $\alpha \preccurlyeq_C^\oplus \beta$ and $\alpha \succcurlyeq_C^\oplus \beta$, i.e., $\bigoplus_{c \in C} c(\alpha) = \bigoplus_{c \in C} c(\beta)$. For $C = \emptyset$, $\alpha \equiv_C^\oplus \beta$ for all $\alpha, \beta \in \mathcal{A}$.

Definition 2.2 (HCLP models)**.** *An HCLP model H for an HCLP structure $\langle \mathcal{A}, \oplus, \mathcal{C} \rangle$ is an ordered partition of a subset $\sigma(H) \subseteq \mathcal{C}$ of evaluations. We write H as sequence $(C_1, \dots, C_k)$, where the (possibly empty) sets $\emptyset \subseteq C_1, \dots, C_k \subseteq \mathcal{C}$ are disjoint and $\bigcup_{i=1,\dots,k} C_k = \sigma(H)$. We say C_i is the i-th level set in H. We denote the empty HCLP model with $\sigma(H) = \emptyset$ by $H = ()$.*

An HCLP model can be viewed as a hierarchy on the evaluation functions. For HCLP model $H = (C_1, \dots, C_k)$ the level set C_1 contains the most important evaluation functions; the level set C_2 contains the second most important evaluation functions and so on. Evaluations $\mathcal{C} \setminus \sigma(H)$ that are not included in the HCLP model are irrelevant for rating the alternatives. Accordingly, we compare two alternatives first on a combination by $\oplus$ of the most important evaluation functions; only if these are equal do we compare them on the combination of the next most important evaluations. Hence, each HCLP model $H = (C_1, \dots, C_k)$ implies a weak order $\preccurlyeq_H^\oplus$ on the alternatives that is a lexicographic order on combinations of evaluations. More specifically, for two alternatives $\alpha, \beta \in \mathcal{A}$, $\alpha \preccurlyeq_H^\oplus \beta$ if and only if

(I) for all $i = 1, \dots, k$, $\alpha \equiv_{C_i}^\oplus \beta$; or

(II) there exists some $i \in \{1, \dots, k\}$ such that $\alpha \prec_{C_i}^\oplus \beta$ and for all $1 \leq j < i$, $\alpha \equiv_{C_j}^\oplus \beta$.

Similarly to $\prec_C^\oplus$ and $\equiv_C^\oplus$, we define the strict weak order $\prec_H^\oplus$ and the equivalence relation $\equiv_H^\oplus$. For $\alpha, \beta \in \mathcal{A}$, $\alpha \prec_H^\oplus \beta$ if and only if $\alpha \preccurlyeq_H^\oplus \beta$ and $\alpha \not\succcurlyeq_H^\oplus \beta$ (i.e., condition (II) holds). Analogue, $\alpha \equiv_H^\oplus \beta$ if and only if $\alpha \preccurlyeq_H^\oplus \beta$ and $\alpha \succcurlyeq_H^\oplus \beta$ (i.e., condition (I) holds). Note, that by these definitions level sets of HCLP models can be empty, but empty level sets do not affect the relations $\preccurlyeq_H^\oplus$, $\prec_H^\oplus$ and $\equiv_H^\oplus$ or any statements based on these relations.

In this paper, we consider special classes of HCLP models where the sizes of the level sets are bounded by some constant. The class $\mathcal{C}(t)$ is defined to be the set of HCLP models with level sets that contain at most t evaluations, i.e., if $H = (C_1, \dots, C_k)$ is in $\mathcal{C}(t)$ then $|C_i| \leq t$ for all $i = 1, \dots, k$. Note, that the model classes satisfy the relation $\mathcal{C}(s) \subseteq \mathcal{C}(t)$ for $s \leq t$. Class $\mathcal{C}(1)$ contains standard lexicographic models.

Example. *Consider the choice of alternatives Apple Pie (AP), Chocolate Cake (CC) and Ice Cream (IC). The desserts are rated by the evaluation functions: calories (c), sugar (s), and fat (f). The values of $\mathcal{C} = \{c, s, f\}$ are percentages of the recommended daily intake as shown in Table 1. Here, 0 is the best possible value, meaning 0% sugar, calories or fat of the recommended daily intake is contained in the dessert. Let $\oplus$ be the standard addition on $\mathbb{Q}$.*

	AP	CC	IC
c	10	13	11
s	23	23	16
f	20	17	24
$f \oplus s$	43	40	40

Table 1: Values of $c, s, f, f \oplus s$ evaluated on AP, CC, IC.

The HCLP model $H = (\{f, s\}, \{c\})$ consists of the pair of most important evaluations f and s followed by the singleton next most important evaluation c. Hence, H is in $\mathcal{C}(2)$ but not in $\mathcal{C}(1)$. Also, H implies the relations:
IC $\prec_H^\oplus$ CC, since $(f(IC) \oplus s(IC), c(IC)) = (40, 11) <_{lex} (40, 13) = (f(CC) \oplus s(CC), c(CC))$. Similarly, IC $\prec_H^\oplus$ AP and CC $\prec_H^\oplus$ AP.

2.2 Preference Consistency

In the following, we first describe the language of preference statements considered in this paper and then define PCP. For this, we define $\mathcal{L}^{\mathcal{A}}_{\leq}$ to be the set of non-strict preference statements $\alpha \leq \beta$ (meaning α is preferred to β) on alternatives $\alpha, \beta \in \mathcal{A}$. Similarly, $\mathcal{L}^{\mathcal{A}}_{<}$ is the set of strict preference statements $\alpha < \beta$ (meaning α is strictly preferred to β) for $\alpha, \beta \in \mathcal{A}$. Let $\mathcal{L}^{\mathcal{A}} = \mathcal{L}^{\mathcal{A}}_{\leq} \cup \mathcal{L}^{\mathcal{A}}_{<}$. We write a preference statement $\varphi \in \mathcal{L}^{\mathcal{A}}_{\leq}$ as $\alpha_\varphi \leq \beta_\varphi$, and $\varphi \in \mathcal{L}^{\mathcal{A}}_{<}$ as $\alpha_\varphi < \beta_\varphi$ for $\alpha_\varphi, \beta_\varphi \in \mathcal{A}$. We denote the non-strict version of preference statements $\Gamma \subseteq \mathcal{L}^{\mathcal{A}}$ by $\Gamma^{(\leq)}$, i.e., $\Gamma^{(\leq)} = \{\alpha_\varphi \leq \beta_\varphi \mid \varphi \in \Gamma\}$.

Definition 2.3 (Satisfaction of Preference Statements)**.** *Let H be an HCLP model for HCLP structure $\langle \mathcal{A}, \oplus, \mathcal{C} \rangle$. We say H satisfies a preference statement $\varphi \in \mathcal{L}^{\mathcal{A}}$, denoted $H \models^{\oplus} \varphi$, if $\alpha_\varphi \preccurlyeq^{\oplus}_{H} \beta_\varphi$ for $\varphi \in \mathcal{L}^{\mathcal{A}}_{\leq}$; or $\alpha_\varphi \prec^{\oplus}_{H} \beta_\varphi$ for $\varphi \in \mathcal{L}^{\mathcal{A}}_{<}$. Furthermore, H satisfies a set of preference statements $\Gamma \subseteq \mathcal{L}^{\mathcal{A}}$, denoted $H \models^{\oplus} \Gamma$, if H satisfies all preference statements in Γ, i.e., $H \models^{\oplus} \varphi$ for all $\varphi \in \Gamma$.*

Definition 2.4 (Consistency)**.** *Let $\langle \mathcal{A}, \oplus, \mathcal{C} \rangle$ be an HCLP structure and $t \in \{1, \ldots, |\mathcal{C}|\}$ a constant. We say $\Gamma \subseteq \mathcal{L}^{\mathcal{A}}$ is $\mathcal{C}(t)$-consistent, if there exists an HCLP model $H \in \mathcal{C}(t)$ such that $H \models^{\oplus} \Gamma$.*

It is easy to show that Γ is $\mathcal{C}(t)$-consistent if Γ is $\mathcal{C}(s)$-consistent for some $s < t$. Also, the empty model $H = ()$ always satisfies non-strict statements, but never satisfies strict statements. Thus, $\Gamma \subseteq \mathcal{L}^{\mathcal{A}}_{\leq}$ is always consistent.

Example (continued)**.** *Consider preference statements:*

(1) I strictly prefer ice cream to apple pie (IC < AP).

(2) I prefer chocolate cake to apple pie (CC ≤ AP).

Then (1) is a strict preference statement, (2) is a non-strict preference statement. As before, HCLP model $H = (\{c, s\}, \{f\})$ satisfies IC $\prec^{\oplus}_{H}$ AP and CC $\prec^{\oplus}_{H}$ AP and so CC $\preceq^{\oplus}_{H}$ AP. Thus, (1) and (2) together are consistent.

We can now formulate the Preference Consistency and Deduction (decision) Problems for classes $\mathcal{C}(t)$.

$\mathcal{C}(t)$ **Preference Consistency Problem (**$\mathcal{C}(t)$**-PCP):** Given an HCLP structure $\langle \mathcal{A}, \oplus, \mathcal{C} \rangle$, a constant $t \in \{1, \ldots, |\mathcal{C}|\}$ and a set of preference statements $\Gamma \subseteq \mathcal{L}^{\mathcal{A}}$. Is Γ $\mathcal{C}(t)$-consistent?

$\mathcal{C}(t)$ **Preference Deduction Problem (**$\mathcal{C}(t)$**-PDP):** Given an HCLP structure $\langle \mathcal{A}, \oplus, \mathcal{C} \rangle$, a constant $t \in \{1, \ldots, |\mathcal{C}|\}$, some preference statements $\Gamma \subseteq \mathcal{L}^{\mathcal{A}}$ and $\varphi \in \mathcal{L}^{\mathcal{A}} \setminus \Gamma$. Does $H \models^{\oplus} \varphi$ hold for all $H \in \mathcal{C}(t)$ with $H \models^{\oplus} \Gamma$?

Wilson et al. [2015] (in their Proposition 1) show that $\mathcal{C}(t)$-PCP and $\mathcal{C}(t)$-PDP are mutually expressive, i.e., PCP can be solved by solving PDP and vice versa:

Proposition 2.5. *$H \models^{\oplus} \varphi$ for all $H \in \mathcal{C}(t)$ with $H \models^{\oplus} \Gamma$ if and only if $\Gamma \cup \{\neg\varphi\}$ is $\mathcal{C}(t)$-inconsistent, where $\neg\varphi = \beta_\varphi < \alpha_\varphi$ for $\varphi \in \mathcal{L}^{\mathcal{A}}_{\leq}$, or $\neg\varphi = \beta_\varphi \leq \alpha_\varphi$ for $\varphi \in \mathcal{L}^{\mathcal{A}}_{<}$.*

They also established that $\mathcal{C}(t)$-PDP is coNP-complete for any $t \geq 2$ (even if $\oplus$ is strictly monotonic). Thus $\mathcal{C}(t)$-PCP is NP-complete. A greedy algorithm can solve the special cases $\mathcal{C}(1)$-PCP and $\mathcal{C}(1)$-PDP in time $O(|\mathcal{C}| \cdot |\Gamma|)$.

3 MILP Formulation

We describe a Mixed Integer Linear Programming (MILP) formulation for $\mathcal{C}(t)$-PCP with HCLP structure $\langle \mathcal{A}, \oplus, \mathcal{C} \rangle$ and preference statements $\Gamma \subseteq \mathcal{L}^{\mathcal{A}}$, where evaluations $\mathcal{C}$ are integral, $\oplus$ is the standard addition on integers and $n = |\mathcal{C}|$.

Assigning Evaluations to Level Sets: We introduce a matrix of Boolean variables $Y \in \{0,1\}^{n \times n}$ such that $y_{i,j} = 1$ if and only if evaluation i is included in the j-th level set of the solution HCLP model. For $\mathcal{C}(t)$-PCP, every evaluation is contained in at most one level set and the cardinality of the level sets is bounded by t.

$$\sum_{j=1}^{n} y_{i,j} \leq 1 \quad \text{and} \quad \sum_{j=1}^{n} y_{j,i} \leq t \;\; \forall i = 1, \ldots, n. \qquad (1)$$

Maintaining Values of $\oplus$-combined Level Sets: The matrix of integer variables $X \in \mathbb{Q}^{n \times |\Gamma|}$ contains the values of the combined evaluation functions in the level sets for the alternatives of the preference statements. Thus, $x_{i,\varphi} = \bigoplus_{c \in C_i} c(\alpha_\varphi) - \bigoplus_{c \in C_i} c(\beta_\varphi)$ maintains the degree of support/opposition of statement φ in the i-th level set C_i.

$$\sum_{i=1}^{n} y_{i,j}(c(\alpha_\varphi) - c(\beta_\varphi)) = x_{j,\varphi} \;\; \forall j = 1, \ldots, n, \forall \varphi \in \Gamma. \qquad (2)$$

We define the bounds $M_\varphi \geq x_{j,\varphi} \geq m_\varphi$ for all $x_{j,\varphi}$ by

$$m_\varphi = \min_{E \subseteq \mathcal{C}} \sum_{c \in E} c(\alpha_\varphi) - c(\beta_\varphi) = \sum_{c \in \mathcal{C}, c(\alpha_\varphi) < c(\beta_\varphi)} c(\alpha_\varphi) - c(\beta_\varphi),$$

$$M_\varphi = \max_{E \subseteq \mathcal{C}} \sum_{c \in E} c(\alpha_\varphi) - c(\beta_\varphi) = \sum_{c \in \mathcal{C}, c(\alpha_\varphi) > c(\beta_\varphi)} c(\alpha_\varphi) - c(\beta_\varphi).$$

Maintaining the Sign of Level Sets (Supporting, Opposing and Indifferent): The Boolean variables $s^{<0}_{j,\varphi}$, $s^{>0}_{j,\varphi}$ and $s^{=0}_{j,\varphi}$ express the sign for $x_{j,\varphi}$. This is, $s^{<0}_{j,\varphi} = 1$ if and only if $x_{j,\varphi} < 0$, $s^{>0}_{j,\varphi} = 1$ if and only if $x_{j,\varphi} > 0$, and $s^{=0}_{j,\varphi} = 1$ if and only if $x_{j,\varphi} = 0$. Since exactly one of the relations holds,

$$s^{<0}_{j,\varphi} + s^{>0}_{j,\varphi} + s^{=0}_{j,\varphi} = 1 \;\; \forall j = 1, \ldots, n, \forall \varphi \in \Gamma. \qquad (3)$$

To enforce the equivalences, we make use of the bounds M_φ and m_φ and the integrity of the evaluations. In particular, we use that the lowest positive value $x_{j,\varphi}$ can take is 1 and the highest negative value is -1.

For the implication $s^{<0}_{j,\varphi} = 1 \Rightarrow x_{j,\varphi} < 0$, we formulate:

$$x_{j,\varphi} + s^{<0}_{j,\varphi}(M_\varphi + 1) \leq M_\varphi \;\; \forall j = 1, \ldots, n, \forall \varphi \in \Gamma. \qquad (4)$$

For the implication $s^{>0}_{j,\varphi} = 1 \Rightarrow x_{j,\varphi} > 0$, we formulate:

$$x_{j,\varphi} + s^{>0}_{j,\varphi}(m_\varphi - 1) \geq m_\varphi \;\; \forall j = 1, \ldots, n, \forall \varphi \in \Gamma. \qquad (5)$$

Finally, we enforce $s^{=0}_{j,\varphi} = 1 \Rightarrow x_{j,\varphi} = 0$ by

$$x_{j,\varphi} - (1 - s^{=0}_{j,\varphi})m_\varphi \geq 0 \;\; \forall j = 1, \ldots, n, \forall \varphi \in \Gamma \text{ and} \qquad (6)$$

$$x_{j,\varphi} - (1 - s^{=0}_{j,\varphi})M_\varphi \leq 0 \;\; \forall j = 1, \ldots, n, \forall \varphi \in \Gamma. \qquad (7)$$

The equivalences follow from (3) together with (4)-(7).

Satisfaction of Strict and Non-strict Statements: Following the definition of $\preccurlyeq^{\oplus}_{H}$, the HCLP model corresponding to the variable assignments of Y satisfies a non-strict statement φ in Γ if and only if

(I′) for all $i = 1, \dots, n$, $s^{=0}_{i,\varphi} = 1$; or

(II′) there exists some $i \in \{1, \dots, n\}$ such that $s^{<0}_{i,\varphi} = 1$ and for all $1 \leq j < i$, $s^{=0}_{j,\varphi} = 1$.

Analogously, a strict statements φ in Γ is satisfied if and only if (II′) holds. It is easy to check that conditions (I′) or (II′) holding for all $\varphi \in \Gamma$ is equivalent to

$$\sum_{j=1}^{i-1} s^{<0}_{j,\varphi} \geq s^{>0}_{i,\varphi} \ \ \forall i = 1, \dots, n, \forall \varphi \in \Gamma. \tag{8}$$

Inequality (8) yields the satisfaction of $\Gamma^{(\leq)}$. We enforce satisfaction of all strict statements in Γ, by

$$\sum_{j=1}^{n} s^{<0}_{j,\varphi} \geq 1 \ \ \forall \varphi \in \Gamma \cap \mathcal{L}^{\mathcal{A}}_{<}. \tag{9}$$

The constraints (1)-(9) form a rather simple MILP formulation for $\mathcal{C}(t)$-PCP. Constraints (3)-(9) could be replaced by sums with extreme weights to enforce a lexicographic order on the level sets. However, these inequalities lead to numerical difficulties for the solver. Also, decision variables $y_{i,j}$ could be substituted by $y'_{i,j}$ such that $y'_{i,j} = 1$ if and only if i is included in a level set $\geq j$. Variables $s^{<0}_{j,\varphi}, s^{>0}_{j,\varphi}, s^{=0}_{j,\varphi} \in \{0, 1\}$ might be replaceable by a variable $s_{j,\varphi} \in \{0, 1, 2\}$. However, since our MILP is a satisfaction problem, not an optimization problem, it is not clear whether any of these measures improve the formulation. After trying various Constraint Programming models with set or binary variables, different versions of constraints and different search heuristics, the MILP formulation using inequalities (1)-(9) seemed most promising.

4 Recursive Algorithms

In the following, we describe two recursive search algorithms for $\mathcal{C}(t)$-PCP. The algorithms are based on properties of PCP that can be used to prune the search space. Both try to construct a Γ-satisfying HCLP model by sequentially adding new level sets to the model that do not oppose any preference statement that is not strictly satisfied so far. Thus, during the algorithm the current model always satisfies $\Gamma^{(\leq)}$, the non-strict version of Γ. We backtrack when the current model cannot be extended further and the model does not satisfy all strict preference statements. The approaches aim to reduce the number of $\Gamma^{(\leq)}$-satisfying HCLP models constructed by the algorithm. In particular, they try to identify and ignore level sets which cannot lead to a Γ-satisfying HCLP model although not opposing the preference statements.

Utilising Sequences of Singleton Level Sets: The first approach is based on the idea of including as many singleton level sets as possible. This seems computationally less challenging since a $\Gamma^{(\leq)}$- satisfying sequence of singleton level sets that is maximal in the number of level sets can be found in time $O(|\mathcal{C}| \cdot |\Gamma|)$ [Wilson *et al.*, 2015]. Remember, that $\Gamma^{(\leq)}$ is defined to be the non-strict version of Γ, i.e., $\Gamma^{(\leq)} = \{\alpha_\varphi \leq \beta_\varphi \mid \varphi \in \Gamma\}$. In the following we show that for strictly monotonic operators $\oplus$ the recursive search algorithm never needs to backtrack over the choice of such singleton sequences. We first establish the following property for strictly monotonic operators $\oplus$ which can be shown by a short technical proof.

Lemma 4.1. *Let $\oplus$ be a strictly monotonic operator and let $X, Y \subseteq \mathcal{C}$ be sets of evaluation functions with $X \subseteq Y$. Let $\alpha, \beta \in \mathcal{A}$ be alternatives such that X is indifferent under α and β, i.e., $\alpha \equiv^{\oplus}_{X} \beta$. Then $\alpha \prec^{\oplus}_{Y} \beta$ if and only if $\alpha \prec^{\oplus}_{Y \setminus X} \beta$. Hence, $\alpha \equiv^{\oplus}_{Y} \beta$ if and only if $\alpha \equiv^{\oplus}_{Y \setminus X} \beta$.*

We define the (non-commutative) combination of two HCLP models $H = (C_1, \dots, C_l)$ and $H' = (C'_1, \dots, C'_k)$ in $C(t)$ as $H \circ H' = (C_1, \dots, C_l, (C'_1 \setminus \sigma_H), \dots, (C'_k \setminus \sigma_H))$, where $\sigma_H = \bigcup_{i=1,\dots,l} C_i$. It is easy to see that $H \circ H'$ is an HCLP model in $C(t)$ as well. The following proposition shows how the satisfaction of preference statements Γ from H' persists under combination with sequences of singleton level sets that only satisfy $\Gamma^{(\leq)}$.

Proposition 4.2. *Let $(\langle \mathcal{A}, \oplus, \mathcal{C} \rangle, \Gamma)$ be an instance of $C(t)$-PCP. If $H = (c_1, \dots, c_l)$ is a $\Gamma^{(\leq)}$- satisfying model in $\mathcal{C}(1)$ and $H' = (C'_1, \dots, C'_k)$ is a Γ-satisfying model in $C(t)$, then $H \circ H'$ is a Γ-satisfying model in $C(t)$.*

Proof. We show, that $H \circ H'$ satisfies $\Gamma^{(\leq)}$ and strictly satisfies the preference statements that H' strictly satisfies. Hence, $H \circ H'$ is a Γ-satisfying HCLP model in $C(t)$.

Recall that a preference statement φ is strictly satisfied when there exists a level set C supporting φ, i.e., $\alpha_\varphi \prec_C \beta_\varphi$, and all preceding level sets C' are indifferent under φ, i.e., $\alpha_\varphi \equiv_{C'} \beta_\varphi$. Hence, the preference statements in Γ that are strictly satisfied by $H = (c_1, \dots, c_l)$ are also strictly satisfied by $H \circ H' = (c_1, \dots, c_l, (C'_1 \setminus \sigma_H), \dots, (C'_k \setminus \sigma_H))$. Let Γ' be the set of remaining preference statements that are not strictly satisfied by H. Since H satisfies $\Gamma^{(\leq)}$, H is indifferent under all statements in $\varphi \in \Gamma'$, i.e., $c_i(\alpha_\varphi) = c_i(\beta_\varphi)$ for all $1 \leq i \leq l$. Consider an arbitrary level set C in H' and a preference statement $\varphi \in \Gamma'$. Repeatedly applying Lemma 4.1 for the singleton level sets in $\sigma_H \cap C$ yields: $\alpha_\varphi \sim^{\oplus}_{C} \beta_\varphi$ if and only if $\alpha_\varphi \sim^{\oplus}_{C \setminus \sigma_H} \beta_\varphi$, where $\sim$ is one of the relations $\prec$, $\equiv$ or $\succ$. Thus, the level sets $C'_i \setminus \sigma_H$ in $H \circ H'$ have the same relation towards statements $\varphi \in \Gamma'$ as the level sets C'_i in H'. Since the initial singleton sequence in $H \circ H'$ is indifferent under preference statements $\varphi \in \Gamma'$, $H \circ H'$ satisfies φ if and only if H' satisfies φ. Also, all statements $\Gamma \setminus \Gamma'$ are strictly satisfied by $H \circ H'$. Hence, $H \circ H'$ satisfies $\Gamma^{(\leq)}$ and strictly satisfies all statements in Γ that H' strictly satisfies. □

Proposition 4.2 immediately leads to the next result.

Proposition 4.3. *Let H be a $\Gamma^{(\leq)}$-satisfying model in $C(1)$ that consists of a maximal number of singleton level sets. If Γ is $C(t)$-consistent, then there exists a Γ-satisfying model in $C(t)$ with H as initial sequence.*

Based on Proposition 4.3, we describe the algorithm PC-check that solves $C(t)$-PCP by trying to construct a Γ-satisfying HCLP model. This method is summarised in the algorithm below disregarding the framed parts. After finding an initial singleton sequence $(c_1, \dots, c_k)$ that is maximal while satisfying $\Gamma^{(\leq)}$ (in time $O(|\mathcal{C}| \cdot |\Gamma|)$ by a greedy algorithm [Wilson *et al.*, 2015]), we consider every possible (not opposing) level set C of size $2 \leq |C| \leq t$. Let Γ' be the set of preference statements in Γ that are not strictly satisfied by $H = (c_1, \ldots, c_k, C)$. We try to extend the sequence H by another Γ'-satisfying HCLP model. We construct this extending model by recursively calling the method for the subproblem with statements Γ' and evaluations $\mathcal{C}' = \mathcal{C} - \{c_1, \ldots, c_k\} - C$. If no such extension exists (that satisfies Γ), we backtrack over the last chosen level set C and try a new level set. Note, that by Proposition 4.3, we never have to backtrack over the choice of singleton level sets, which can be a significant advantage over solving the MILP model. As soon as the currently considered sequence in the algorithm satisfies Γ we stop and return the sequence, i.e., showing that the instance is $C(t)$-consistent. If no Γ-satisfying sequence can be found after exploiting all possible ($\Gamma^{(\leq)}$-satisfying) HCLP model, we stop and return that the instance is $C(t)$-inconsistent.

Maintaining Conflicting Sets: In the following, we extend the algorithm PC-check($\mathcal{C}, \Gamma, \oplus, t$) by maintaining conflicting sets that cannot be contained in the later level sets, and thus reduce the number of backtracks. Proposition 4.4 shows that the satisfaction of Γ persists for an HCLP model H' when combining with an HCLP model H that extends an initial sequence of level sets of H' by one more level set and only satisfies $\Gamma^{(\leq)}$.

Proposition 4.4. *Let* $(\langle \mathcal{A}, \oplus, \mathcal{C} \rangle, \Gamma)$ *be an instance of* $C(t)$*-PCP that is* $C(t)$*-consistent. Let* $H = (C_1, \ldots, C_k, B)$ *be a* $\Gamma^{(\leq)}$*-satisfying HCLP model in* $C(t)$ *and let* $H' = (C_1, \ldots, C_k, C_{k+1}, \ldots, C_l)$ *be a* Γ*-satisfying HCLP model in* $C(t)$ *with* $B \subseteq C_j$ *for some* $k + 1 \leq j \leq l$*. Then* $H \circ H' = (C_1, \ldots, C_k, B, C_{k+1}, \ldots, (C_j \setminus B), \ldots, C_l)$ *is a* Γ*-satisfying HCLP model in* $C(t)$*.*

Proposition 4.4 is proved in a similar way as Proposition 4.2 using Lemma 4.1. Negating the statement yields: If there exists no Γ-satisfying HCLP model in $C(t)$ that extends H, then there exists no Γ-satisfying HCLP model in $C(t)$ that extends $(C_1, \ldots, C_k)$ and contains B in some level set. In the second recursive approach, extending the algorithm below by framed parts, we maintain such *conflicting* sets B. When the algorithm backtracks because no Γ-satisfying extension of the current HCLP model can be found, we mark the last added level set C to be a conflicting set if $|C| \leq s$. Then, a new next level set is chosen that does not contain any conflicting set. Although reducing the search space, a potentially exponentially large list of conflicting sets is maintained, where s regulates the space complexity.

```
Algorithm: PC-check($\mathcal{C}, \Gamma, \oplus, t, \boxed{\mathcal{S} = \emptyset, s}$)
$H \leftarrow (c_1, \ldots, c_k)$ some $\Gamma^{(\leq)}$-satisfying singleton sequence with $k$ maximal
if $H \models \Gamma$ then return $H$
for all $C \subseteq \mathcal{C} - \{c_1, \ldots, c_k\}$ with $2 \leq |C| \leq t$
$\boxed{\text{such that there exists no } S \in \mathcal{S} \text{ with } S \subseteq C}$
and $\alpha_\varphi \preccurlyeq^{\oplus}_{C} \beta_\varphi$ for all $\varphi$ with $\alpha_\varphi \equiv^{\oplus}_{H} \beta_\varphi$ do
    $H \leftarrow (c_1, \ldots, c_k, C)$
    if $H \models \Gamma$ then return $H$
    else if $H \models \Gamma^{(\leq)}$ then
        $\Gamma' = \{\varphi \in \Gamma \mid \alpha_\varphi \equiv^{\oplus}_{H} \beta_\varphi\}, \mathcal{C}' = \mathcal{C} - \sigma(H)$
        $H \leftarrow (c_1, \ldots, c_k, C,$PC-check$(\mathcal{C}', \Gamma', \oplus, t, \boxed{\mathcal{S}, s}))$
        if $H \models \Gamma$ then return $H$ $\boxed{\textbf{else } \mathcal{S} \leftarrow \mathcal{S} \cup C}$
return Inconsistent
```

Example. *Consider a* $\mathcal{C}(3)$*-PCP instance with the following evaluation functions* $c_1, \ldots, c_5$ *and statements on* $\alpha, \beta, \gamma, \delta$*.*

	α	$\leq$	β	$\leq$	γ	$<$	δ
c_1	1		0		0		0
c_2	0		2		2		2
c_3	1		1		0		1
c_4	0		2		1		1
c_5	2		0		1		0

Let $\oplus$ *be the standard addition on* $\mathbb{Z}$*. Suppose, in the first step PC-check finds the maximal singleton sequence* (c_2, c_1) *(which cannot be extended by any other evaluation without violating* $\Gamma^{(\leq)}$*). Then the algorithm will in turn consider sets* $\{c_3, c_4\}$*,* $\{c_3, c_5\}$*,* $\{c_4, c_5\}$ *and* $\{c_3, c_4, c_5\}$*. The sequences* $c_2, c_1, \{c_3, c_4\}$ *and* $c_2, c_1, \{c_4, c_5\}$ *violate* $\Gamma^{(\leq)}$*. The sequence* $c_2, c_1, \{c_3, c_5\}$ *satisfies* $\Gamma^{(\leq)}$ *but cannot be extended to satisfy* Γ*. In PC-check(*$\mathcal{C}, \Gamma, \oplus, t, \mathcal{S}, s$*), the set* $\{c_3, c_5\}$ *is added to the conflicting sets* $\mathcal{S}$ *and thus the set* $\{c_3, c_4, c_5\}$ *does not have to be checked (by Proposition 4.4). PC-check(*$\mathcal{C}, \Gamma, \oplus, t$*) finds that* $c_2, c_1, \{c_3, c_4, c_5\}$ *violates* $\Gamma^{(\leq)}$*. Thus none of the possible sets leads to a* Γ*-satisfying sequence and "Inconsistent" is returned. Note, that PC-check does not have to backtrack over the choice of evaluations in the initial singleton set sequence* c_2, c_1 *(by Proposition 4.3).*

5 Experimental Runtime Comparisons

In our experiments, we compare the approaches from Section 3 and 4 for solving PCP by their running time. Here, the MILP formulation functions as a baseline and is expected to be outperformed by the two recursive approaches as they directly exploit the problem structure to perform less backtracks in a way that is not recognized by CPLEX. Note, that it is not obvious how to incorporate the pruning of the search space, that is preformed by the recursive algorithms, in a MILP model in form of constraints or heuristics. We investigate the degree of improvement of the recursive algorithms towards the rather simple MILP formulation and the relation of the recursive algorithms towards each other. Though PC-check($\mathcal{C}, \Gamma, \oplus, t, \mathcal{S}, s$) prunes the search space further than PC-check($\mathcal{C}, \Gamma, \oplus, t$), the list of maintained conflicting sets can grow exponentially large. Thus, it is not obvious if maintaining conflicting sets is advantageous.

Instances: For our experiments we considered different instance sizes, to observe the effect on the running time by varying the number of evaluations n and the number of preference statements g. For the lack of real world data, we generated 50 instances uniformly at random with evaluation functions with domains $\{0, 1, 2, 3, 4, 5\}$ for each of the problem sizes $n, g \in \{10, 15, \ldots, 50\}$ where we fix the number of alternatives that the preference statements are based on to $m = 25$. Note, that not all alternatives generated are involved in preference statements. Thus, m has no direct influence on the size of the search space or the running time. For time reasons, some experiments were not conducted for all instance sizes.

Implementation: We implemented all three approaches in Java Version 1.8 using the IBM ILOG CPLEX (version 12.6.2) library for the MILP formulation. All experiments were conducted independently on a 2.66Ghz quad-core processor with 12GB memory.

We choose $\oplus$ as the standard addition on the integers. To reduce the number of experiments, we allow the cardinality bound on the level sets to be $t = n$, the number of evaluations, and fix the cardinality bound on the maintained conflicting sets to $s = 5$ (which gives the bound $|\mathcal{S}| \leq \binom{n}{s} \leq \binom{50}{5} = 2118760$). Since $\mathcal{C}(k') \subseteq \mathcal{C}(k)$ for all $k' < k$, we expect the running times to be lower for smaller t. Also, $\mathcal{C}(k) = \mathcal{C}(n)$ for all $k \geq n$, i.e., the running times are the same for bigger t.

Experimental Results: As expected, solving the MILP formulation of PCP (as presented in Section 3) by the CPLEX solver is much slower than by the two recursive algorithms PC-check (as presented in Section 4), see Table 2. However, it is remarkable how quickly the ratio between the mean times of solving the MILP and PC-check grows with the number of statements and evaluations in the instances.

$g =$		$n = 10$	$n = 15$	$n = 20$
10	PC-check($\mathcal{C}, \Gamma, \oplus, t$)	0.011	0.01	0.04
	PC-check($\mathcal{C}, \Gamma, \oplus, t, \mathcal{S}, s$)	0.003	0.01	0.03
	MILP	0.22	10.53	149.81
	minimal ratio: MILP/PC-check	20	1053	3745.25
15	PC-check($\mathcal{C}, \Gamma, \oplus, t$)	0.003	0.01	0.29
	PC-check($\mathcal{C}, \Gamma, \oplus, t, \mathcal{S}, s$)	0.001	0.01	0.28
	MILP	0.22	220.28	>41472
	minimal ratio: MILP/PC-check	73.33	22028	>143006.89

Table 2: Mean times in seconds to solve PCP with the MILP formulation over 25 and with PC-check over 50 instances.

The two algorithms PC-check($\mathcal{C}, \Gamma, \oplus, t$) and PC-check($\mathcal{C}, \Gamma, \oplus, t, \mathcal{S}, s$) show similar behavior of running times for different instance sizes. Figure 1 shows some of the running times for PC-check($\mathcal{C}, \Gamma, \oplus, t$) and PC-check($\mathcal{C}, \Gamma, \oplus, t, \mathcal{S}, s$). Here, we can see that the running times increase with the number of evaluations n and the number of statements g.

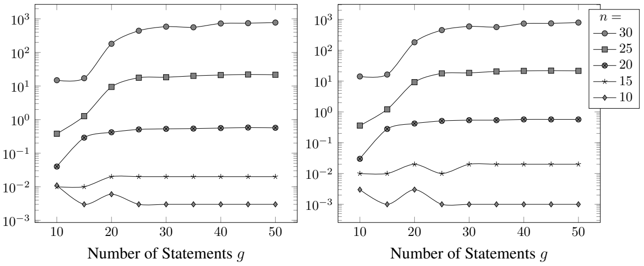

Figure 1: Mean times in seconds of PC-check($\mathcal{C}, \Gamma, \oplus, t$) (left) and PC-check($\mathcal{C}, \Gamma, \oplus, t, \mathcal{S}, s$) (right).

A detailed analysis shows that in 57% of the measured instance sizes PC-check($\mathcal{C}, \Gamma, \oplus, t$) is slower than PC-check($\mathcal{C}, \Gamma, \oplus, t, \mathcal{S}, s$). The ratios of the mean running times of the two algorithms demonstrate that PC-check($\mathcal{C}, \Gamma, \oplus, t$) is at most 3.67 times slower than PC-check($\mathcal{C}, \Gamma, \oplus, t, \mathcal{S}, s$) and PC-check($\mathcal{C}, \Gamma, \oplus, t, \mathcal{S}, s$) is at most 1.03 times slower than PC-check($\mathcal{C}, \Gamma, \oplus, t$). This insignificant difference between the running times of the two algorithms means that for these instances reducing the number of backtracks by ruling out conflicting sets is not worth the effort and space needed to maintain the list of conflicting sets (of size < 5).

To find an explanation for the behavior of the running times as shown in Figure 1, we observed the occurrence of instances that have solutions in $\mathcal{C}(1)$, in $\mathcal{C}(t)$ with $t > 1$, or are inconsistent. The whole search space must be explored until deciding inconsistency for all t, which can lead to high running times. PC-check solves $\mathcal{C}(1)$ instances in polynomial time. The instance distribution may suggest that the running times go up with the number of inconsistent instances, e.g., see Figure 2.

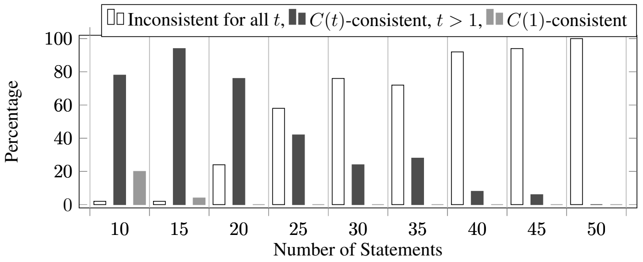

Figure 2: Percentages of instance classes for the tested random instances with $n = 30$ (top) and $n = 35$ (bottom).

6 Conclusion

Exploiting the theoretical results on properties of consistent instances developed in Section 4 allow the algorithms PC-check to prune the search space much further than a MILP solver could do for the MILP formulation given in Section 3. The experimental results confirm, that the algorithms PC-check are solving the instances faster than CPLEX. Even more, the ratios between the mean solving times of the MILP and PC-check increase extremely quickly with the number of evaluations and statements. It is not obvious how the pruning rules of the PC-check algorithms can be incorporated in the MILP formulation as constraints.

There is no significant difference between the mean running times of the two recursive algorithms PC-check on the tested instances. Thus, in PC-check($\mathcal{C}, \Gamma, \oplus, t, \mathcal{S} = \emptyset, s$), the effort of maintaining a list $\mathcal{S}$ of (possibly exponentially many) conflicting sets to prune the search space further, is not paying off. An analysis of the instance consistency types indicates that the running times for algorithms PC-check might increase with the number of inconsistent instances.

A further analysis could involve the size of the search space, i.e., counting the number of $\Gamma^{(\leq)}$-satisfying HCLP models and the number of HCLP models that were actually considered during the search. In future work, we will try using a relaxation of a MILP formulation as a fast check for inconsistency within PC-check($\mathcal{C}, \Gamma, \oplus, t$). If the relaxation shows that the current subproblem is inconsistent, we can avoid another (time consuming) recursive call.

Acknowledgment

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

[Booth *et al.*, 2010] R. Booth, Y. Chevaleyre, J. Lang, J. Mengin, and C. Sombattheera. Learning conditionally lexicographic preference relations. In *ECAI '10, Lisbon, Portugal, August 16-20, 2010, Proceedings*, pages 269–274, 2010.

[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.

[Bräuning and Hüllermeier, 2012] M. Bräuning and E. Hüllermeier. Learning conditional lexicographic preference trees. In *Preference Learning (PL-12), ECAI '12 workshop, Montpellier, France, August 28th, 2012, Proceedings*, 2012.

[Bridge and Ricci, 2007] D. Bridge and F. Ricci. Supporting product selection with query editing recommendations. In *RecSys '07, Minneapolis, MN, USA, October 19-20, 2007, Proceedings*, pages 65–72. ACM, 2007.

[Dombi *et al.*, 2007] J. Dombi, C. Imreh, and N. Vincze. Learning lexicographic orders. *European Journal of Operational Research*, 183(2):748–756, 2007.

[Figueira *et al.*, 2005] J. Figueira, S. Greco, and M. Ehrgott. *Multiple Criteria Decision Analysis—State of the Art Surveys*. Springer International Series in Operations Research and Management Science Volume 76, 2005.

[Flach and Matsubara, 2007] P. A. Flach and E.T. Matsubara. A simple lexicographic ranker and probability estimator. In *ECML '07, Warsaw, Poland, September 17-21, 2007, Proceedings*, pages 575–582, 2007.

[Fürnkranz and Hüllermeier, 2010] J. Fürnkranz and E. Hüllermeier. *Preference Learning*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[George *et al.*, 2015] A.-M. George, A. Razak, and N. Wilson. The comparison of multi-objective preference inference based on lexicographic and weighted average models. In *ICTAI '15, Vietri sul Mare, Italy, November 9-11, 2015, Proceedings*, pages 88–95, 2015.

[George *et al.*, 2016] A.-M. George, N. Wilson, and B. O'Sullivan. *Towards Fast Algorithms for the Preference Consistency Problem Based on Hierarchical Models (extended version of IJCAI-2016)*. http://ucc.insight-centre.org/nwilson/PCPalgorithmsProofs.pdf, 2016.

[Liu and Truszczynski, 2015] Xudong Liu and Miroslaw Truszczynski. Learning partial lexicographic preference trees over combinatorial domains. In *AAAI '15, Austin Texas, USA, January 2530, 2015, Proceedings*, pages 1539–1545. AAAI Press, 2015.

[Marinescu *et al.*, 2013] R. Marinescu, A. Razak, and N. Wilson. Multi-objective constraint optimization with tradeoffs. In *CP '13, Uppsala, Sweden, September 16-20, 2013, Proceedings*, pages 497–512, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[Trabelsi *et al.*, 2011] W. Trabelsi, N. Wilson, D. Bridge, and F. Ricci. Preference dominance reasoning for conversational recommender systems: a comparison between a comparative preferences and a sum of weights approach. *International Journal on Artificial Intelligence Tools*, 20(4):591–616, 2011.

[Wilson and Borning, 1993] M. Wilson and A. Borning. Hierarchical constraint logic programming. *The Journal of Logic Programming*, 16(34):277–318, 1993.

[Wilson *et al.*, 2015] N. Wilson, A.-M. George, and B. O'Sullivan. *Computation and Complexity of Preference Inference Based on Hierarchical Models (extended version of IJCAI-2015)*. http://ucc.insight-centre.org/nwilson/PrefInfHCLPproofs.pdf), 2015.

[Wilson, 2014] N. Wilson. Preference inference based on lexicographic models. In *ECAI '14, 18-22 August 2014, Prague, Czech Republic - Including PAIS '14*, pages 921–926, 2014.