

# Preference Trees over Combinatorial Domains

Xudong Liu

Ph.D. Candidate  
Department of Computer Science  
College of Engineering  
University of Kentucky  
Lexington, KY, USA

# Preferences Are Ubiquitous

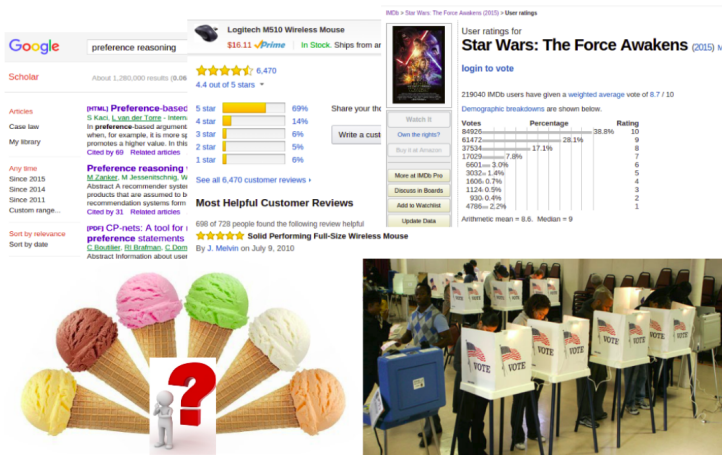


Figure : Preferences of different forms

# Describing Preferences



Figure : How to express it?

- ① On scale of 0 to 99, how will I rate these two cars?
  - I give Car1 44 points and Car2 78 points; thus, I prefer Car2 to Car1.
- ② Which one to me is better than the other?
  - I prefer Car1 to Car2. (Strict preference)
  - I like Car1 and Car2 equally. (Indifference/Equivalence)
  - I cannot decide. (Incomparability)

# Describing Preferences



Figure : How to express it?

- ① On scale of 0 to 99, how will I rate these two cars? (**Quantitative**)
  - I give Car1 44 points and Car2 78 points; thus, I prefer Car2 to Car1.
- ② Which one to me is better than the other? (**Qualitative**)
  - I prefer Car1 to Car2. (Strict preference)
  - I like Car1 and Car2 equally. (Indifference/Equivalence)
  - I don't know. (Incomparability)

## Binary Relations

Let  $O$  be a set of elements. A *binary relation*  $\preceq$  over  $O$  is a collection of ordered pairs of elements in  $O$ ; that is,

$$\preceq \subseteq O \times O.$$

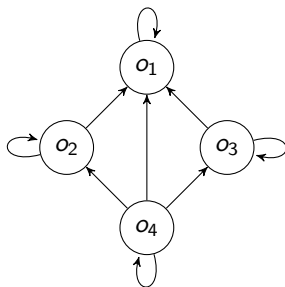
Properties of binary relations:

- ① Reflexivity:  $\forall o \in O, o \preceq o$ .
- ② Irreflexivity:  $\forall o \in O, o \not\preceq o$ .
- ③ Totality:  $\forall o_1, o_2, o_1 \preceq o_2$  or  $o_2 \preceq o_1$ .
- ④ Transitivity:  $\forall o_1, o_2, o_3$ , if  $o_1 \preceq o_2$  and  $o_2 \preceq o_3$ , then  $o_1 \preceq o_3$ .
- ⑤ Symmetricity:  $\forall o_1, o_2$ , if  $o_1 \preceq o_2$ , then  $o_2 \preceq o_1$ .
- ⑥ Antisymmetricity:  $\forall o_1, o_2$ , if  $o_1 \preceq o_2$  and  $o_2 \preceq o_1$ , then  $o_1 = o_2$ .

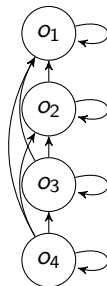
# Relations and Orderings

## Binary Relations

$\preceq$  is a *partial preorder* if it is reflexive and transitive, a *total preorder* if it is a partial preorder and total, a *partial order* if it is a partial preorder and antisymmetric, and a *total order* if it is a partial order and total.



(a) partial (pre)order

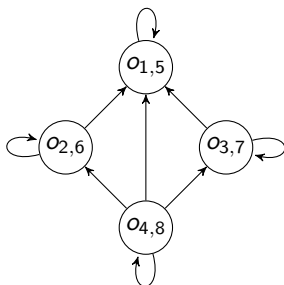


(b) total (pre)order

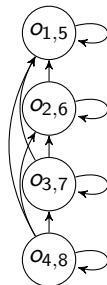
# Relations and Orderings

## Binary Relations

$\preceq$  is a *partial preorder* if it is reflexive and transitive, a *total preorder* if it is a partial preorder and total, a *partial order* if it is a partial preorder and antisymmetric, and a *total order* if it is a partial order and total.



(a) partial preorder

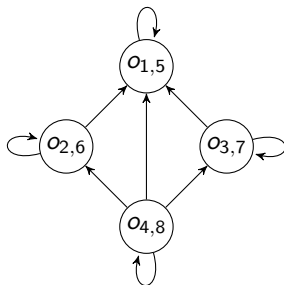


(b) total preorder

# Relations and Orderings

## Binary Relations

Let  $\preceq$  be a preference relation that is a partial preorder over  $O$ . We say that  $o_2$  is *weakly preferred* to  $o_1$  if  $o_1 \preceq o_2$ , that  $o_2$  is *strictly preferred* ( $\prec$ ) to  $o_1$  if  $o_1 \preceq o_2$  and  $o_2 \not\preceq o_1$ , that  $o_1$  is *indifferent* ( $\approx$ ) from  $o_2$  if  $o_1 \preceq o_2$  and  $o_2 \preceq o_1$ , and that  $o_1$  is *incomparable* ( $\sim$ ) with  $o_2$  if  $o_1 \not\preceq o_2$  and  $o_2 \not\preceq o_1$ .



(a) partial preorder

$$o_1 \preceq o_5,$$

$$o_4 \prec o_2,$$

$$o_4 \approx o_8,$$

$$o_6 \sim o_7.$$

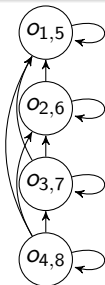
(b) preferences



# Relations and Orderings

## Binary Relations

Let  $\preceq$  be a preference relation that is a partial preorder over  $O$ . We say that  $o_2$  is *weakly preferred* to  $o_1$  if  $o_1 \preceq o_2$ , that  $o_2$  is *strictly preferred* ( $\prec$ ) to  $o_1$  if  $o_1 \preceq o_2$  and  $o_2 \not\preceq o_1$ , that  $o_1$  is *indifferent* ( $\approx$ ) from  $o_2$  if  $o_1 \preceq o_2$  and  $o_2 \preceq o_1$ , and that  $o_1$  is *incomparable* ( $\sim$ ) with  $o_2$  if  $o_1 \not\preceq o_2$  and  $o_2 \not\preceq o_1$ .



(a) total preorder

$$o_1 \preceq o_5,$$

$$o_4 \prec o_2,$$

$$o_4 \approx o_8,$$

(b) preferences

## Combinatorial Domains

Let  $V$  be a finite set of variables  $\{X_1, \dots, X_p\}$ ,  $D$  a set of finite domains  $\{Dom(X_1), \dots, Dom(X_p)\}$  for each variable  $X_i$ . A *combinatorial domain*  $CD(V)$  is a set of *outcomes* described by combinations of values from  $Dom(X_i)$ :

$$CD(V) = \prod_{X_i \in V} Dom(X_i).$$

# Combinatorial Domains: Example

Domain of cars over set  $V$  of  $p$  binary variables:

① **BodyType**: {mvan, sedan}.

② **Capacity**: {5, 7m}.

③ **Color**: {blue, grey}.

⋮

$$CD(V) = \underbrace{\{\langle \text{sedan}, 4, \text{blue}, \dots \rangle, \langle \text{mvan}, 6\text{m}, \text{grey}, \dots \rangle, \dots\}}_{2^p \text{ outcomes, too many!}}.$$

# Computational Complexity

- ①  $P$  ( $NP$ ): decision problems solvable by a deterministic (nondeterministic, resp.) TM in poly time in the size of the input.
  - We typically believe that  $P \subset NP$ .
- ②  $coNP$ : problems whose complements are in  $NP$ .
- ③  $\Delta_2^P$ :  $P^{NP}$ .
- ④  $\Sigma_2^P$ :  $NP^{NP}$ .
- ⑤  $PSPACE$ : decision problems solvable by a TM in poly space in the size of the input.
- ⑥ A decision problem  $L$  is  $C$ -hard if  $L' \leq_p L$  for every  $L'$  in class  $C$ .
- ⑦ A decision problem  $L$  is  $C$ -complete if  $L$  is in class  $C$  and  $L$  is  $C$ -hard.

# Computational Complexity

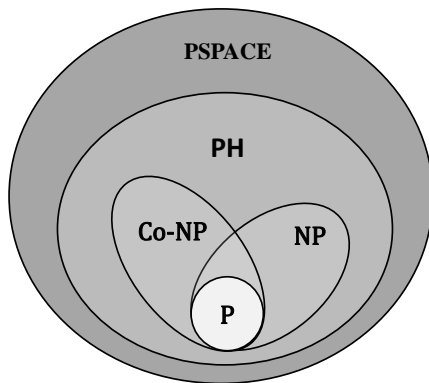


Figure : Computational complexity diagram

# Combinatorial Domains: Example

Domain of cars (cf. the Car Evaluation Dataset<sup>1</sup>)

- ① **BodyType**: {mvan, sedan, sport, suv}.
- ② **Capacity**: {2, 5, 7m}.
- ③ **Color**: {black, blue, grey, red, white}.
- ④ **LuggageSize**: {big, med, small}.
- ⑤ **Make**: {bmw, ford, honda, and vw}.
- ⑥ **Price**: {low, med, high, vhigh}.
- ⑦ **Safety**: {low, med, high}.

---

<sup>1</sup><http://www.cs.uky.edu/~liu/preflearnlib.php>, slightly adapted in the talk.

# Qualitative Preferences

Individual:

Car1



<mvan, 6m, grey, big, honda, med, med>



Car2



<sedan, 4, blue, med, vw, med, med>

Figure : Dominance Testing

# Qualitative Preferences

Collective:



Figure : Social Choice and Welfare



Q: How do we represent qualitative preferences over combinatorial domains?

- ① Answer-Set Optimization Theories<sup>2</sup>
- ② Ceteris Paribus Networks (e.g., CP-nets<sup>3</sup>, TCP-nets<sup>4</sup>, CI-nets<sup>5</sup>)
- ③ Conditional Preference Theories<sup>6</sup>

---

<sup>2</sup>Gerhard Brewka, Ilkka Niemelä, and Mirosław Truszczyński. "Answer Set Optimization". In: IJCAI. 2003

<sup>3</sup>C. Boutilier et al. "CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements". In: Journal of Artificial Intelligence Research (2004)

<sup>4</sup>Ronen I. Brafman and Carmel Domshlak. "Introducing Variable Importance Tradeoffs into CP-Nets". In: UAI. 2002

<sup>5</sup>Sylvain Bouveret, Ulle Endriss, and Jérôme Lang. "Conditional importance networks: A graphical language for representing ordinal, monotonic preferences over sets of goods". In: (2009)

<sup>6</sup>Nic Wilson. "Extending CP-Nets with Stronger Conditional Preference Statements". In: AAAI-04. 2004

Q: How do we represent qualitative preferences over combinatorial domains?

- ④ Preference Trees (e.g., **LP-trees**<sup>16,20</sup>, **CLP-trees**<sup>17</sup>, **PLP-trees**<sup>18</sup>, **P-trees**<sup>11,21</sup>)

---

<sup>7</sup>Richard Booth et al. "Learning conditionally lexicographic preference relations". In: ECAI. 2010

<sup>8</sup>Xudong Liu and Mirosław Truszczyński. "Aggregating Conditionally Lexicographic Preferences Using Answer Set Programming Solvers". In: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT). 2013

<sup>9</sup>Michael Bräuning and H Eyke. "Learning Conditional Lexicographic Preference Trees". In: Preference learning: problems and applications in AI (2012)

<sup>10</sup>Xudong Liu and Mirosław Truszczyński. "Learning Partial Lexicographic Preference Trees over Combinatorial Domains". In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI). 2015

<sup>11</sup>Niall M Fraser. "Ordinal preference representations". In: Theory and Decision (1994)

<sup>12</sup>Xudong Liu and Mirosław Truszczyński. "Preference Trees: A Language for Representing and Reasoning about Qualitative Preferences". In: Proceedings of the 8th Multidisciplinary Workshop on Advances in Preference Handling (MPREF). 2014

Q: How do we learn predictive qualitative preference models over combinatorial domains?

- ① Ceteris Paribus Networks (e.g., CP-nets<sup>13,14,15</sup>)
- ② Preference Trees (e.g., LP-trees<sup>16</sup>, CLP-trees<sup>17</sup>, **PLP-trees**<sup>18</sup>)
- ③ **Preference Forests**<sup>19</sup>

---

<sup>13</sup>Jérôme Lang and Jérôme Mengin. "The complexity of learning separable ceteris paribus preferences". In: (2009)

<sup>14</sup>Frédéric Koriche and Bruno Zanuttini. "Learning conditional preference networks". In: Artificial Intelligence (2010)

<sup>15</sup>Yann Chevaleyre et al. "Learning ordinal preferences on multiattribute domains: The case of CP-nets". In: Preference learning. 2011

<sup>16</sup>Richard Booth et al. "Learning conditionally lexicographic preference relations". In: ECAI. 2010

<sup>17</sup>Michael Bräuning and H Eyke. "Learning Conditional Lexicographic Preference Trees". In: Preference learning: problems and applications in AI (2012)

<sup>18</sup>Xudong Liu and Mirosław Truszczyński. "Learning Partial Lexicographic Preference Trees over Combinatorial Domains". In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI). 2015

<sup>19</sup>Xudong Liu and Mirosław Truszczyński. "Learning Preference Trees and Forests". In: IJCAI-16 (In Preparation)

Q: How do we reason about preferences over combinatorial domains?

## ① Preference Optimization<sup>20,21,22</sup>:

- Dominance testing:  $o_1 \succ_P o_2$ ?
- Optimality testing:  $o_1 \succ_P o_2$  for all  $o_2 \neq o_1$ ?
- Optimality computing: what is the optimal outcome wrt  $P$ ?
- Ranking: how are the outcomes ordered wrt  $P$ ?

## ② Preference Misrepresentation<sup>23</sup>:

- Control
- **Manipulation**<sup>24</sup>
- Bribery

---

<sup>20</sup>Xudong Liu and Mirosław Truszczynski. "Aggregating Conditionally Lexicographic Preferences Using Answer Set Programming Solvers". In: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT). 2013

<sup>21</sup>Xudong Liu and Mirosław Truszczynski. "Preference Trees: A Language for Representing and Reasoning about Qualitative Preferences". In: Proceedings of the 8th Multidisciplinary Workshop on Advances in Preference Handling (MPREF). 2014

<sup>22</sup>Xudong Liu and Mirosław Truszczynski. "Reasoning with Preference Trees over Combinatorial Domains". In: Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT). 2015

<sup>23</sup>Felix Brandt, Vincent Conitzer, and Ulle Endriss. "Computational social choice". In: Multiagent systems (2012)

<sup>24</sup>Xudong Liu and Mirosław Truszczynski. "Complexity of Manipulation in Elections Where Votes Are Lexicographic Preference Trees". In: AAMAS-17 (In Preparation)

Q: What fields can we apply preferences to?

- ① Game Theory:
  - **Hedonic games**<sup>25</sup>
- ② Automated Planning and Scheduling:
  - **Trip planning**<sup>26</sup>
- ③ Data-Driven Decision Making:
  - **Predictive decisions**<sup>27</sup>

---

<sup>25</sup>Matthew Spradling et al. "Roles and Teams Hedonic Game". In: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT). 2013

<sup>26</sup>Xudong Liu et al. "On Personalizability and Extensibility of Multi-Modal Trip Planning". In: PARC Symposium. 2015

<sup>27</sup>Xudong Liu and Mirosław Trzuszczynski. "Learning Preference Trees and Forests". In: IJCAI-16 (In Preparation)

- 1 Let  $\mathcal{I} = \{X_1, \dots, X_p\}$  be a set of attributes, and  $D(\mathcal{I}) = \{Dom(X_1), \dots, Dom(X_p)\}$  a set of finite domains for  $\mathcal{I}$ .
- 2 A *literal* is an assignment to an attribute. We denote by  $X_i := x_{i,j}$  the literal that assigns value  $x_{i,j} \in Dom(X_i)$  to  $X_i$ . When no confusion, we write  $x_{i,j}$ , instead of  $X_i := x_{i,j}$ , as a literal. We then denote by  $\mathcal{L} = \{x_{i,j} \in Dom(X_i) : X_i \in \mathcal{I}\}$  the set of literals given  $\mathcal{I}$  and  $D(\mathcal{I})$ .
- 3 The combinatorial domain  $CD(\mathcal{I})$  is defined as earlier.

- 4 A **P-tree**  $T$  over  $CD(\mathcal{I})$  is a binary tree whose nodes, other than the leaves, are labeled with propositional formulas over  $\mathcal{L}$ .
- 5 Given an outcome  $M \in CD(\mathcal{I})$ , the **leaf**  $l_T(M)$  is the leaf reached by traversing the tree  $T$  according to  $M$ . When at a node  $N$  labeled with  $\varphi$ , if  $M \models \varphi$ , we descend to the left child of  $N$ ; otherwise, to the right.
- 6 For  $M, M' \in CD(\mathcal{I})$ , we have  $M \succ_T M'$  if  $l_T(M) \succ_T l_T(M')$ , and  $M \approx_T M'$  if  $l_T(M) = l_T(M')$ . Outcome  $M$  is **optimal** if there exists no  $M'$  such that  $M' \succ_T M$ .

# Example: The Car Domain

- ① **BodyType**( $X_1$ ):  $\{\text{mvan}(x_{1,1}), \text{sedan}(x_{1,2}), \text{sport}(x_{1,3}), \text{suv}(x_{1,4})\}$ .
- ② **Capacity**( $X_2$ ):  $\{2, 5, 7\text{m}\}$ .
- ③ **Color**( $X_3$ ):  $\{\text{black}, \text{blue}, \text{grey}, \text{red}, \text{white}\}$ .
- ④ **LuggageSize**( $X_4$ ):  $\{\text{big}, \text{med}, \text{small}\}$ .
- ⑤ **Make**( $X_5$ ):  $\{\text{bmw}, \text{ford}, \text{honda}, \text{and vw}\}$ .
- ⑥ **Price**( $X_6$ ):  $\{\text{low}, \text{med}, \text{high}, \text{vhigh}\}$ .
- ⑦ **Safety**( $X_7$ ):  $\{\text{low}, \text{med}, \text{high}\}$ .



# Example: Preference Trees over Cars

**BodyType**( $X_1$ ): {mvan( $x_{1,1}$ ), sedan( $x_{1,2}$ ), sport( $x_{1,3}$ ), suv( $x_{1,4}$ )}.

**Color**( $X_3$ ): {black, blue, grey, red, white}.

**Price**( $X_6$ ): {low, med, high, vhigh}.

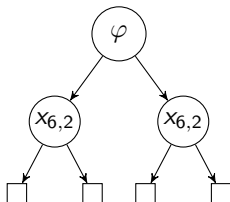


Figure : A P-tree over vacations<sup>28</sup>

---

<sup>28</sup>  $\varphi = (x_{1,1} \wedge x_{3,3}) \vee (x_{1,2} \wedge x_{3,2})$ .

# Example: Preferences over Vacations

**BodyType**( $X_1$ ): {mvan( $x_{1,1}$ ), sedan( $x_{1,2}$ ), sport( $x_{1,3}$ ), suv( $x_{1,4}$ )}.

**Color**( $X_3$ ): {black, blue, grey, red, white}.

**Price**( $X_6$ ): {low, med, high, vhigh}.

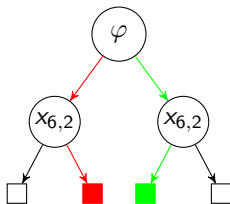


Figure : A P-tree over vacations<sup>28</sup>

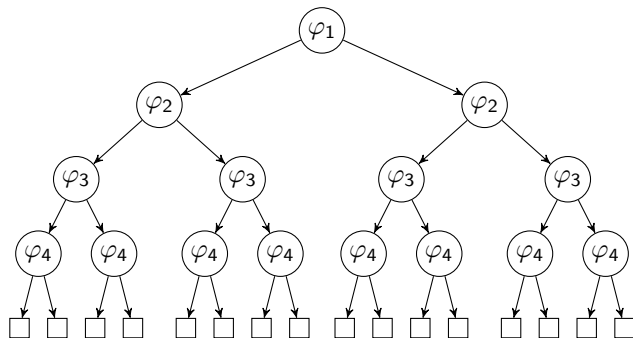
Preferences:

$$\neg x_1 \neg x_2 x_3 \neg x_4 \succ x_1 \neg x_2 x_3 x_4$$

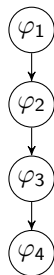
---

<sup>28</sup>  $\varphi = (x_{1,1} \wedge x_{3,3}) \vee (x_{1,2} \wedge x_{3,2})$ .

# Compact Representation of P-trees



(a) Full



(b) Compact

Figure : Compact P-trees

# Compact Representation of P-trees

A *compact P-tree* over  $CD(\mathcal{I})$  is a tree where

- 1 every node is labeled with a Boolean formula over  $\mathcal{I}$ , and
- 2 every non-leaf node  $t$  labeled with  $\varphi$  has either two outgoing edges (Figure (a)), or one outgoing edge pointing left (Figure (b)), right (Figure (c)), or straight-down (Figure (d)).

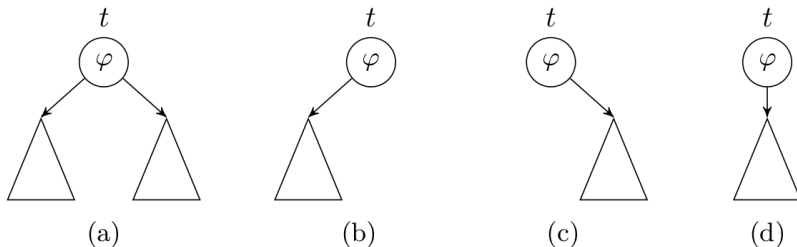
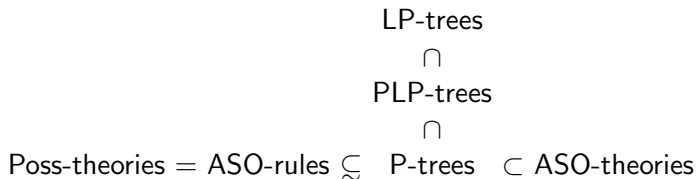


Figure : Compact P-trees

# Relative Expressivity of Preference Languages



# Computational Complexity Results

DOMTEST: is it that  $o \succeq_T o'$  in P-tree  $T$ ?

OPTTEST: is outcome  $o$  optimal w.r.t  $T$ ?

OPTPROP: is there an optimal outcome  $o$  w.r.t  $T$  st  $o \models \alpha$ ?

	DOMTEST	OPTTEST	OPTPROP
LP-tree	P	P	P
ASO-rule/ Poss-theory	P	coNP-c	$\Delta_2^P(P^{NP})$
<b>P-tree</b>	<b>P</b>	<b>coNP-c</b> <sup>29</sup>	$\Delta_2^P(P^{NP})$ -c <sup>30</sup>
ASO-theory	P	coNP-c	$\Sigma_2^P(NP^{NP})$ -c

Figure : Computational complexity results

<sup>29</sup>The complement problem is reduced from the SAT problem.

<sup>30</sup>The problem is reduced from the Maximum Satisfying Assignment (MSA) problem.

# Partial Lexicographic Preference Trees (PLP-Tree)

A *PLP-tree* over  $CD(\mathcal{I})$  is a labeled tree, where

- 1 every node  $t$  is labeled with a attribute  $Attr(t)$  in  $\mathcal{I}$  and a conditional preference table  $CPT(t)$ ,
- 2 every non-leaf node  $t$  has either one unlabeled outgoing edge or multiple outgoing edges labeled, each labeled by some value in  $Dom(Attr(t))$ , and
- 3 every attribute appears *at most* once on every branch.

# Partial Lexicographic Preference Trees (PLP-Tree)

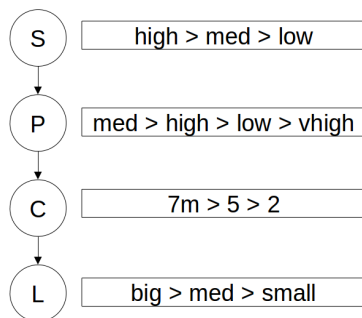


Figure : A UIUP PLP-tree

According to this UIUP PLP-tree, *Car1* is preferred to *Car2*.



# Partial Lexicographic Preference Trees (PLP-Tree)

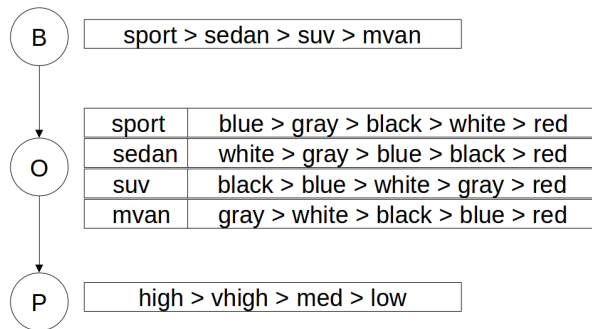


Figure : A UICP PLP-tree

According to this UICP PLP-tree, *Car2* is preferred to *Car1*.

# Partial Lexicographic Preference Trees (PLP-Tree)

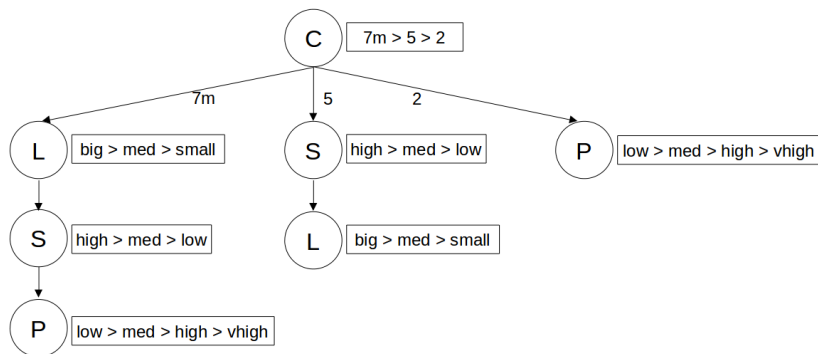


Figure : A CIUP PLP-tree

According to this CICIP PLP-tree, *Car1* is preferred to *Car2*.

# Partial Lexicographic Preference Trees (PLP-Tree)

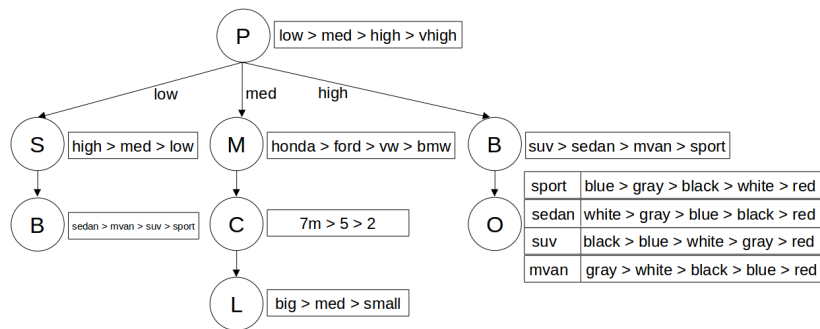


Figure : A CICP PLP-tree

According to this CICP PLP-tree, *Car1* is preferred to *Car2*.

# Complexity Results

	UP	CP
UI	P	P
CI	NPC <sup>31</sup>	P

(a) CONSLearn

	UP	CP
UI	NPC	NPC
CI	NPC	NPC

(b) SMALLLearn

	UP	CP
UI	NPC <sup>32</sup>	NPC
CI	NPC	NPC

(c) MAXLearn

Figure : Complexity results for passive learning problems

<sup>31</sup>Booth et al., *Learning Conditionally Lexicographic Preference Relations*, 2010.

<sup>32</sup>Schmitt and Martignon, *On the Complexity of Learning Lexicographic Strategies*, 2006.

# Experimental Results on Trees

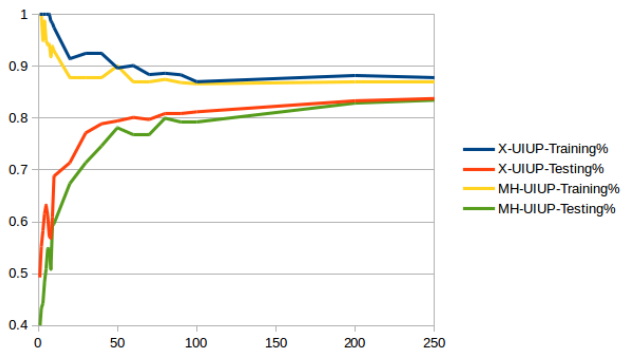


Figure : Learning curve for UIUP using ASP and greedy heuristic

# Experimental Results on Trees

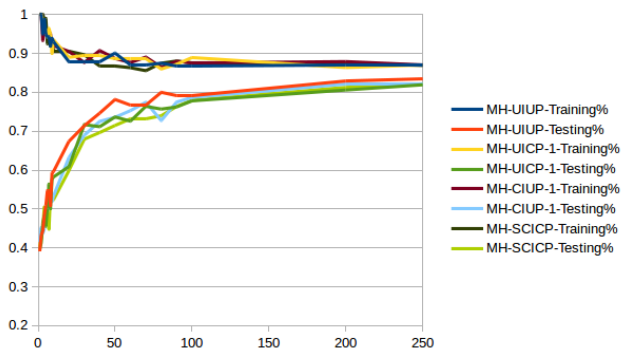


Figure : Learning curve for all four classes using greedy heuristic

- ① A *preference forest*  $F$  is a collection of PLP-trees  $F = \{T_1, \dots, T_n\}$ .
- ② Denote by  $N_F(o_1, o_2) = |\{T \in F : o_1 \succ_T o_2\}|$ .
- ③ Given a preference forest  $F$ , and two outcomes  $o_1$  and  $o_2$ , we say that  $o_1 \succ_F^{Maj} o_2$  iff  $N_F(o_1, o_2) > N_F(o_2, o_1)$ , and that  $o_1 \approx_F^{Maj} o_2$  iff  $N_F(o_1, o_2) = N_F(o_2, o_1)$ .
  - Pro: intuitive, decided in polynomial time.
  - Con: Condorcet paradox.
  - Other aggregating rules: positional scoring rules, Copeland's method, etc.

# Experimental Results on Forests

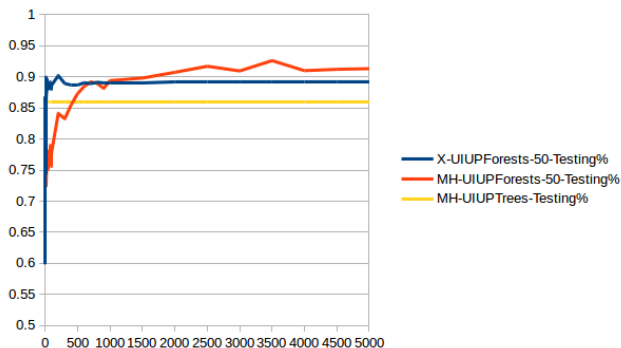


Figure : Learning UIUP using ASP and greedy heuristic



# Experimental Results on Forests

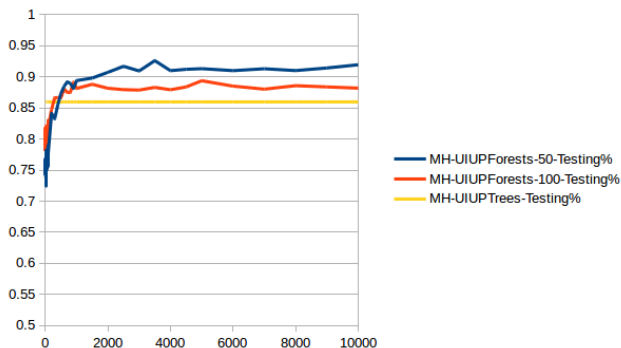


Figure : Learning all four classes using greedy heuristic

# Lexicographic Preference Trees (LP-Trees)

- ① An *LP tree*  $\mathcal{L}$  over  $\mathcal{I} = \{X_1, \dots, X_p\}$  is a (*binary*) tree, where
- each node  $t$  in  $\mathcal{L}$  is labeled by an issue from  $\mathcal{I}$  and with *preference information*, and
  - each issue appears **exactly once** on every path from the root to a leaf.

# Positional Scoring Rules

- $k$ -approval:  $(1, \dots, 1, 0, \dots, 0)$  with  $k$  being the number of 1's and  $m - k$  the number of 0's where  $m = 2^p$ .
- $(k, l)$ -approval:  $(a, \dots, a, b, \dots, b, 0, \dots, 0)$ , where  $a$  and  $b$  are constants ( $a > b$ ) and the numbers of  $a$ 's and  $b$ 's equal to  $k$  and  $l$ , respectively.
- $b$ -Borda:  $(b, b - 1, \dots, 0)$ , where if  $b > m - 1$ ,  $b$ -Borda is reduced to the regular Borda rule with  $(m - 1, m - 2, \dots, 1, 0)$ .

# The Evaluation and Winner Problems

## The Evaluation Problem

Let  $r$  be a positional scoring rule with a scoring vector  $w$ ,  $\mathcal{C}$  a class of LP trees. Given a  $\mathcal{C}$ -profile  $P$  of  $n$  LP trees over  $p$  issues and a positive integer  $R$ , the *evaluation* problem is to decide whether there exists an alternative  $o \in \mathcal{X}$  such that  $s_w(o, P) \geq R$ .

## The Winner Problem

Let  $r$  be a positional scoring rule with a scoring vector  $w$ ,  $\mathcal{C}$  a class of LP trees. Given a  $\mathcal{C}$ -profile  $P$  of  $n$  LP trees over  $p$  issues, the *winner* problem is to compute an alternative  $o \in \mathcal{X}$  with the maximum score  $s_w(o, P)$ .

# Complexity of the Evaluation Problem: $k$ -Approval

	UP	CP
UI	P	P
CI	P	P

(a)  $k = 2^{p-1} \pm f(p)$ ,  $f(p)$  is a poly

	UP	CP
UI	NPC	NPC
CI	NPC	NPC

(b)  $k = 2^{p-c}$ ,  $c > 1$  is a const

Figure :  $k$ -Approval

# Complexity of the Evaluation Problem: $(k, l)$ -Approval

	UP	CP
UI	P	P
CI	P	P

(a)  $k = l = 2^{p-1}$

	UP	CP
UI	NPC	NPC
CI	NPC	NPC

(b)  $k = l = 2^{p-c}$ ,  $c > 1$  is a const

Figure :  $(k, l)$ -Approval <sup>33</sup>

<sup>33</sup> Liu and Truszczynski, *Aggregating Conditionally Lexicographic Preferences Using Answer Set Programming*, ADT, 2013.

# Complexity of the Evaluation Problem: $b$ -Borda

	UP	CP
UI	P	NPC
CI	NPC	NPC

(a)  $b = 2^p - 1$

	UP	CP
UI	NPC	NPC
CI	NPC	NPC

(b)  $b = 2^{p-c} - 1$ ,  $c \geq 1$  is a const

Figure :  $b$ -Borda

# Modeling the Problems in ASP

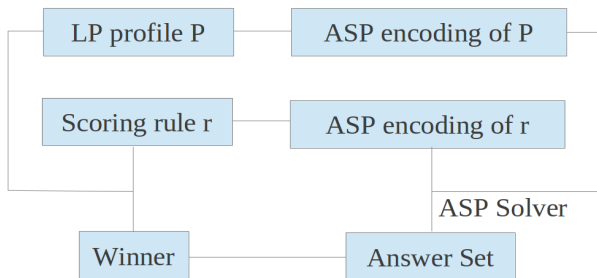


Figure : The winner problem

- Solvers: *clingo*<sup>34</sup>, *clingcon*<sup>35</sup>

<sup>34</sup>M. Gebser et al. "Potassco: The Potsdam Answer Set Solving Collection". In: AI Communications (2011)

<sup>35</sup>Max Ostrowski and Torsten Schaub. "ASP modulo CSP: The clingcon system". In: TPLP (2012)



# Modeling the Problems in W-MAXSAT

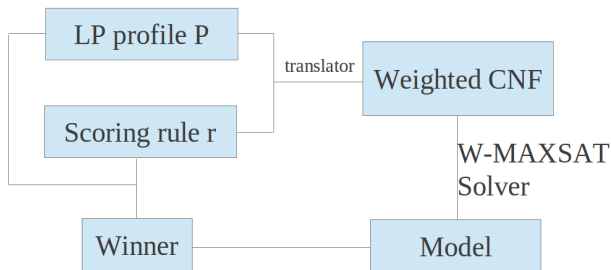


Figure : The winner problem

- Solver: *toulbar*<sup>36</sup>

<sup>36</sup>M Sanchez et al. "Max-CSP competition 2008: toulbar2 solver description". In: the Third International CSP Solver Competition (2008)

# Random LP Profiles

- To experiment with LP profiles, we developed methods to randomly generate *encodings* of a special type of CI-CP LP tree of size linear in the number of issues

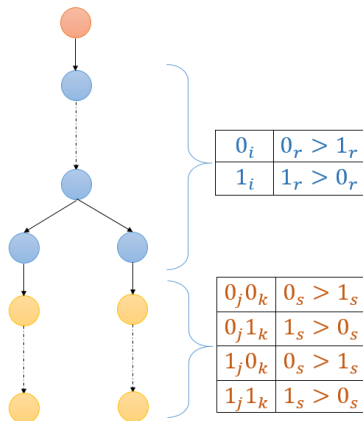
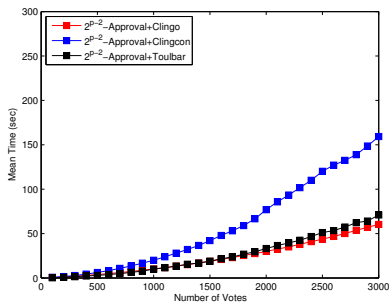
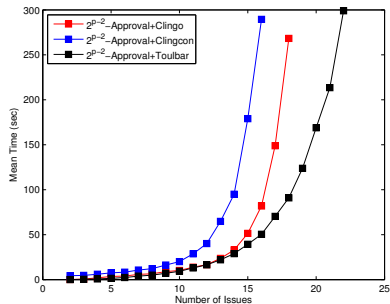


Figure : Random LP tree

# Varying $p$ and $n$ : $2^{p-2}$ -approval



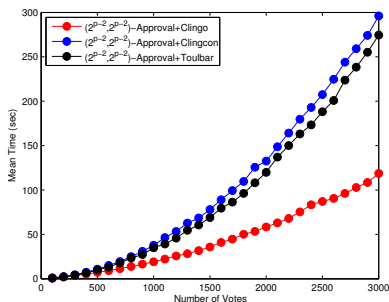
(a) Fixed #issues (10)



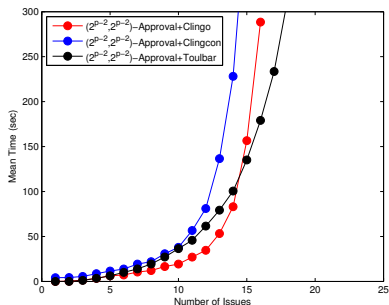
(b) Fixed #votes (1000)

Figure : Solving the winner problem

# Varying $p$ and $n$ : $(2^{p-2}, 2^{p-2})$ -approval <sup>37</sup>



(a) Fixed #issues (10)

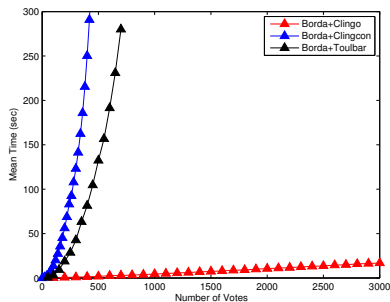


(b) Fixed #votes (1000)

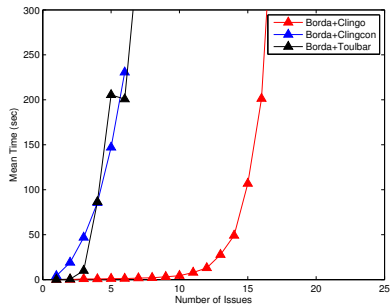
Figure : Solving the winner problem

<sup>37</sup> scoring vector:  $(2, \dots, 2, 1, \dots, 1, 0, \dots, 0)$  with the numbers of 2's and 1's equal to  $2^{p-2}$

# Varying $p$ and $n$ : Borda



(a) Fixed #issues (10)



(b) Fixed #votes (1000)

Figure : Solving the winner problem

# Personalization in Trip Planning

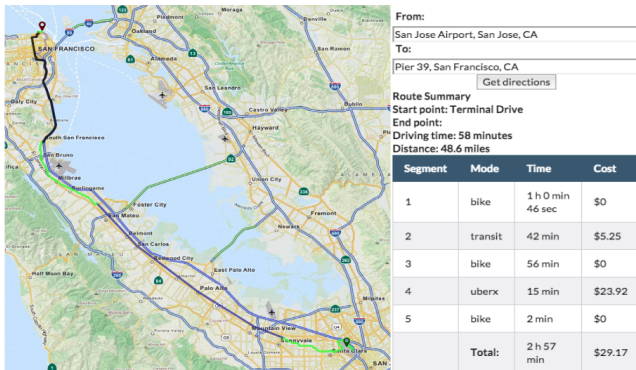
- 1 Important to incorporate user constraints and preferences into trip planning systems.
- 2 Collaboration with experts (in AI, planning, optimization, multi-agent systems) at PARC.
- 3 Developed a hipergraph-based trip planner that accommodates constraints specified as *linear temporal logic* and preferences expressed as *preferential cost function* to compute optimal routes using  $A^*$ <sup>38</sup>.
- 4 Available later for trip planning in the Bay Area, LA, and Denver.

---

<sup>38</sup>Xudong Liu et al. "On Personalizability and Extensibility of Multi-Modal Trip Planning". In: PARC Symposium. 2015

# Personalization in Trip Planning

- 1 From SJC, to Pier 39, Monday, 9am.
- 2 Constraints: never drive a car, and bike for 1 to 2 hours.
- 3 Preferences: bike = public (0.25) > wait(2) > walk(3), and 30\$/hr.



# Data-Driven Preference Engineering

## 1 Recommender Systems<sup>39</sup>:

- 1 Collaborative
- 2 Content-based
- 3 Hybrid

## 2 Machine Learning:

- 1 Supervised learning (e.g., decision trees, random forests)
- 2 Label ranking<sup>40</sup>

## 3 Preference Elicitation (Human-in-the-Loop):

- 1 Context-based

## 4 Preference Learning:

- 1 Conditional Preference Networks, Preference Trees
- 2 Stochastic Models (e.g., Choquet integral<sup>41</sup>, TOPSIS-like models<sup>42</sup>)

---

<sup>39</sup>Gediminas Adomavicius and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions". In: Knowledge and Data Engineering, IEEE Transactions on (2005)

<sup>40</sup>Eyke Hüllermeier et al. "Label ranking by learning pairwise preferences". In: Artificial Intelligence (2008)

<sup>41</sup>Agnes Leroy, Vincent Mousseau, and Marc Pirlot. "Learning the parameters of a multiple criteria sorting method". In: Algorithmic decision theory. 2011

<sup>42</sup>Manish Agarwal, Ali Fallah Tehrani, and Eyke Hüllermeier. "Preference-based Learning of Ideal Solutions in TOPSIS-like Decision Models". In: Journal of Multi-Criteria Decision Analysis (2014)



# Preference Reasoning and Applications

- ① Social Choice and Welfare<sup>43</sup>:
  - ① Voting
  - ② Fair division
  - ③ Strategyproof Social Choice
- ② Automated Planning and Scheduling:
  - ① Travel scheduling
  - ② Manufacturing
  - ③ Traffic control
- ③ Computer Vision and Image Processing:
  - ① Image retrieval
  - ② Image and video understanding

---

<sup>43</sup>Kenneth J Arrow, Amartya Sen, and Kotaro Suzumura. Handbook of Social Choice and Welfare. Vol. 1 & 2. 2010

Thank you!