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

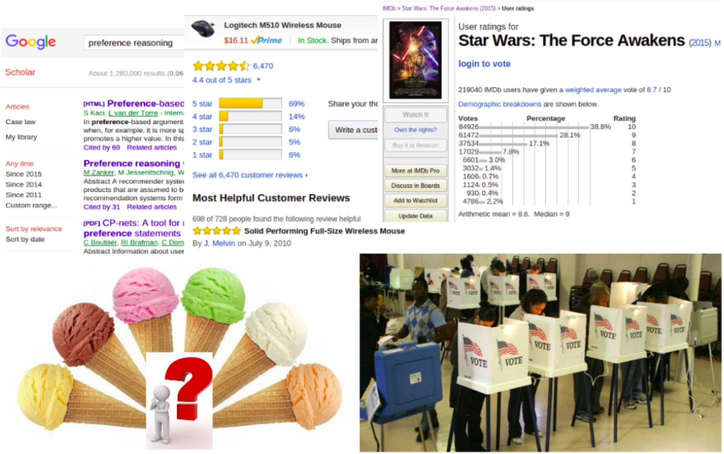


Figure : Preferences of different forms

Describing Preferences

Car1 Car2

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)

Relations and Orderings

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,

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

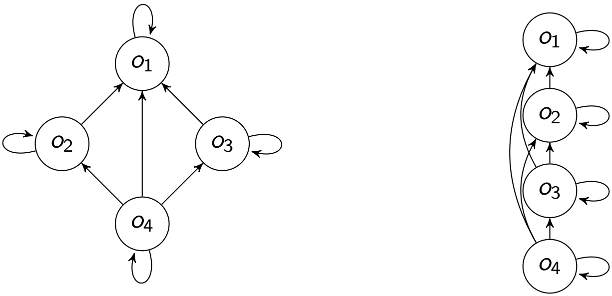
Properties of binary relations:

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.

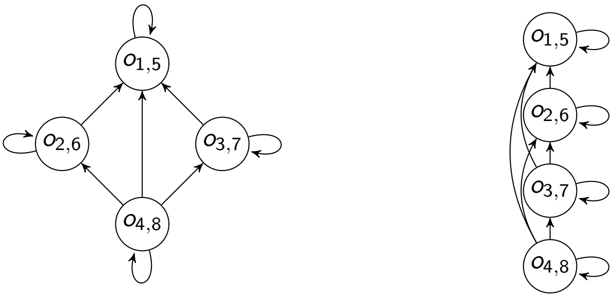


(a) partial (pre)order (b) total (pre)order

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.

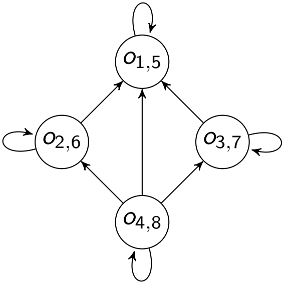


(a) partial preorder

(b) total preorder

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

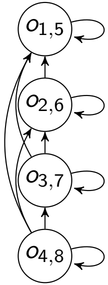
$$o_1 \preceq o_5,$$
$$o_4 \prec o_2,$$
$$o_4 \approx o_8,$$
$$o_6 \sim o_7.$$

(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

Combinatorial Domains

Let V be a finite set of variables $\{X_1, \ldots, X_p\}$, D a set of finite domains $\{Dom(X_1), \ldots, Dom(X_p)\}$ 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$\}$.

⋮

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

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. Δ_2^P: P^{NP}.
4. Σ_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

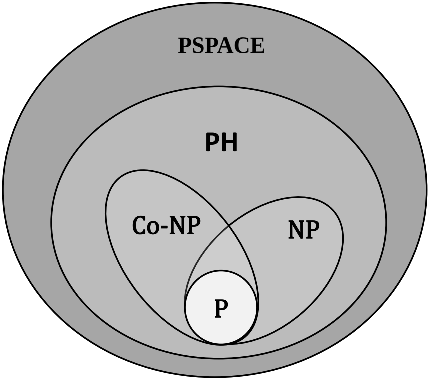


Figure : Computational complexity diagram

Combinatorial Domains: Example

Domain of cars (cf. the Car Evaluation Dataset[1])

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}.

[1] `http://www.cs.uky.edu/~liu/preflearnlib.php`, slightly adapted in the talk.

Individual:

Figure : Dominance Testing

Qualitative Preferences

Collective:

Figure : Social Choice and Welfare

Preference Modeling

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

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

---

[2] Gerhard Brewka, Ilkka Niemelä, and Miroslaw Truszczynski. "Answer Set Optimization". In: IJCAI. 2003

[3] 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)

[4] Ronen I. Brafman and Carmel Domshlak. "Introducing Variable Importance Tradeoffs into CP-Nets". In: UAI. 2002

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

[6] Nic Wilson. "Extending CP-Nets with Stronger Conditional Preference Statements". In: AAAI-04. 2004

Preference Modeling

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

4. Preference Trees (e.g., **LP-trees**[16,20], CLP-trees[17], **PLP-trees**[18], **P-trees**[11,21])

---

[7] Richard Booth et al. "Learning conditionally lexicographic preference relations". In: ECAI. 2010

[8] Xudong Liu and Miroslaw Truszczynski. "Aggregating Conditionally Lexicographic Preferences Using Answer Set Programming Solvers". In: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT). 2013

[9] Michael Bräuning and H Eyke. "Learning Conditional Lexicographic Preference Trees". In: Preference learning: problems and applications in AI (2012)

[10] Xudong Liu and Miroslaw Truszczynski. "Learning Partial Lexicographic Preference Trees over Combinatorial Domains". In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI). 2015

[11] Niall M Fraser. "Ordinal preference representations". In: Theory and Decision (1994)

[12] Xudong Liu and Miroslaw 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

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

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

---

[13] Jérôme Lang and Jérôme Mengin. "The complexity of learning separable ceteris paribus preferences". In: (2009)

[14] Frédéric Koriche and Bruno Zanuttini. "Learning conditional preference networks". In: Artificial Intelligence (2010)

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

[16] Richard Booth et al. "Learning conditionally lexicographic preference relations". In: ECAI. 2010

[17] Michael Bräuning and H Eyke. "Learning Conditional Lexicographic Preference Trees". In: Preference learning: problems and applications in AI (2012)

[18] Xudong Liu and Miroslaw Truszczynski. "Learning Partial Lexicographic Preference Trees over Combinatorial Domains". In: Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI). 2015

[19] Xudong Liu and Miroslaw Truszczynski. "Learning Preference Trees and Forests". In: IJCAI-16 (In Preparation)

Preference Reasoning

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

1. **Preference Optimization**[20,21,22]:
    - 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[23]:
    - Control
    - **Manipulation**[24]
    - Bribery

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

[21] Xudong Liu and Miroslaw 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

[22] Xudong Liu and Miroslaw Truszczynski. "Reasoning with Preference Trees over Combinatorial Domains". In: Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT). 2015

[23] Felix Brandt, Vincent Conitzer, and Ulle Endriss. "Computational social choice". In: Multiagent systems (2012)

[24] Xudong Liu and Miroslaw Truszczynski. "Complexity of Manipulation in Elections Where Votes Are Lexicographic Preference Trees". In: AAMAS-17 (In Preparation)

Preference Applications

Q: What fields can we apply preferences to?

1. Game Theory:
   - **Hedonic games**[25]
2. Automated Planning and Scheduling:
   - **Trip planning**[26]
3. Data-Driven Decision Making:
   - **Predictive decisions**[27]

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

[26] Xudong Liu et al. "On Personalizability and Extensibility of Multi-Modal Trip Planning". In: PARC Symposium. 2015

[27] Xudong Liu and Miroslaw Truszczynski. "Learning Preference Trees and Forests". In: IJCAI-16 (In Preparation)

Preference Trees

- Let $\mathcal{I}$ be a set of binary issues. The **combinatorial domain** $CD(\mathcal{I})$ is the set of *outcomes* represented by complete and consistent sets of literals over $\mathcal{I}$.
- A **P-tree** T over $CD(\mathcal{I})$ is a binary tree whose nodes, other than the leaves, are labeled with propositional formulas over $\mathcal{I}$.
- 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 φ, if $M \models \varphi$, we descend to the left child of N; otherwise, to the right.
- 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 $M' \succ_T M$.

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 φ has either two outgoing edges (Figure (a)), or one outgoing edge pointing left (Figure (b)), right (Figure (c)), or straight-down (Figure (d)).

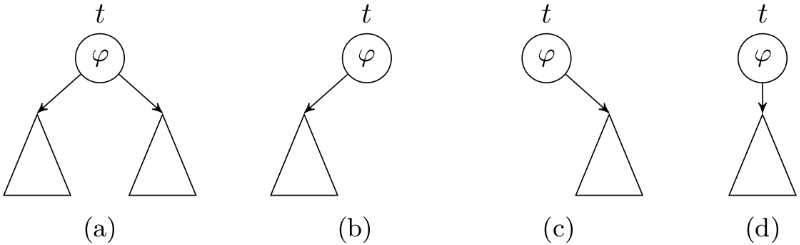


Figure : Compact P-trees

Relative Expressivity of Preference Languages

$$
\begin{array}{ccccccc}
 & & & & \text{LP-trees} & & \\
 & & & & \cap & & \\
 & & & & \text{PLP-trees} & & \\
 & & & & \cap & & \\
\text{Poss-theories} & = & \text{ASO-rules} & \precapprox & \text{P-trees} & \subset & \text{ASO-theories}
\end{array}
$$

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	$\Delta_2^P(P^{NP})$
P-tree	**P**	**coNP-c**[28]	$\boldsymbol{\Delta_2^P(P^{NP})}$**-c**[29]
ASO-theory	P	coNP-c	$\Sigma_2^P(NP^{NP})$-c

Figure : Computational complexity results

[28] The complement problem is reduced from the SAT problem.

[29] 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)

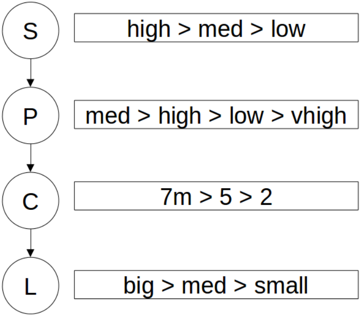


Figure : A UIUP PLP-tree

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

Partial Lexicographic Preference Trees (PLP-Tree)

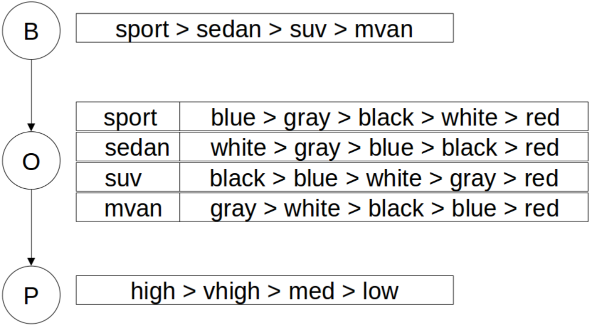


Figure : A UICP PLP-tree

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

Partial Lexicographic Preference Trees (PLP-Tree)

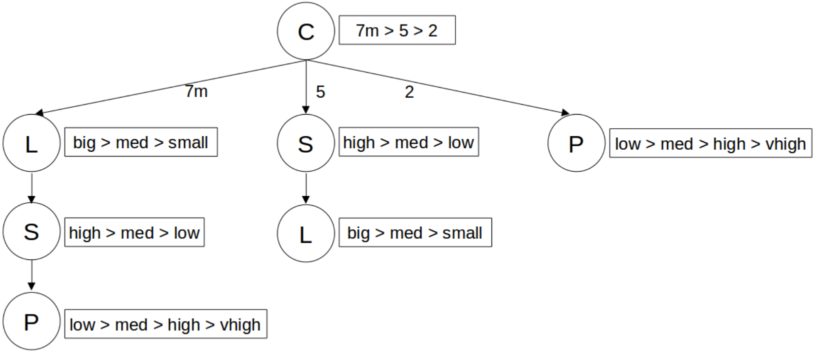


Figure : A CIUP PLP-tree

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

Partial Lexicographic Preference Trees (PLP-Tree)

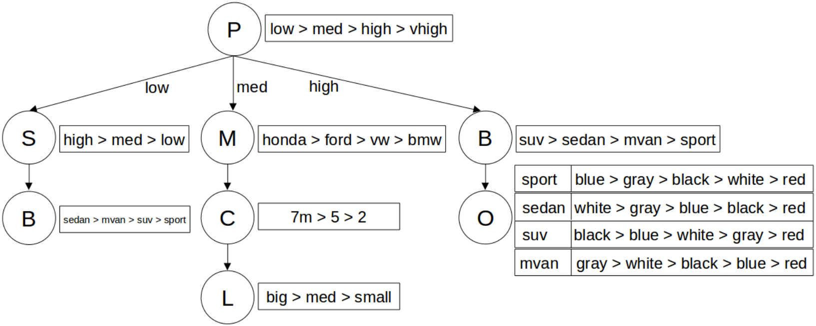


Figure : A CICP PLP-tree

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

Complexity Results

	UP	CP
UI	P	P
CI	NPC[30]	P

(a) ConsLearn

	UP	CP
UI	NPC	NPC
CI	NPC	NPC

(b) SmallLearn

	UP	CP
UI	NPC[31]	NPC
CI	NPC	NPC

(c) MaxLearn

Figure : Complexity results for passive learning problems

[30] Booth et al., *Learning Conditionally Lexicographic Preference Relations*, 2010.

[31] Schmitt and Martignon, *On the Complexity of Learning Lexicographic Strategies*, 2006.

Experimental Results on Trees

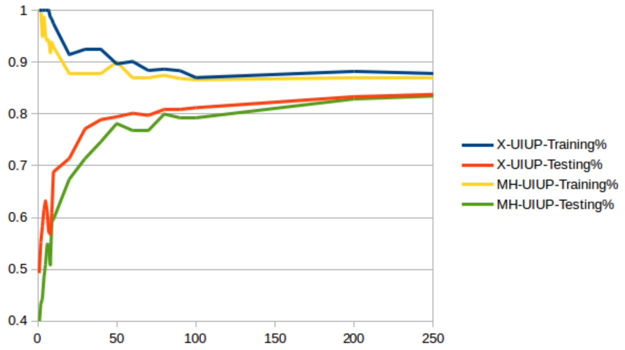


Figure : Learning curve for UIUP using ASP and greedy heuristic

Experimental Results on Trees

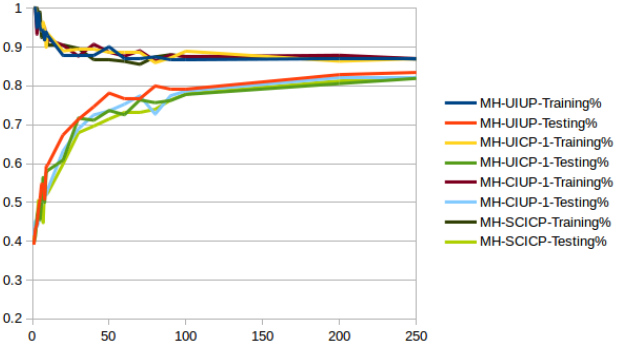


Figure : Learning curve for all four classes using greedy heuristic

Preference Forests

1. A *preference forest* F is a collection of PLP-trees $F = \{T_1, \ldots, T_n\}$.
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

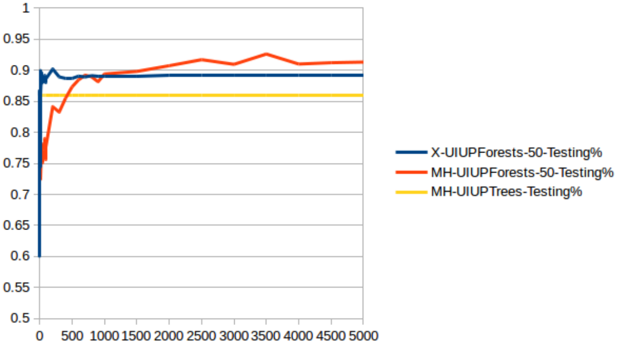


Figure : Learning UIUP using ASP and greedy heuristic

Experimental Results on Forests

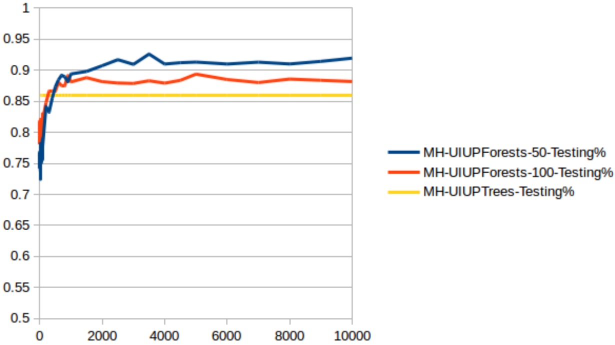


Figure : Learning all four classes using greedy heuristic

Lexicographic Preference Trees (LP-Trees)

1. An *LP tree* $\mathcal{L}$ over $\mathcal{I} = \{X_1, \ldots, X_p\}$ 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: $(1, \ldots, 1, 0, \ldots, 0)$ with k being the number of 1's and $m - k$ the number of 0's where $m = 2^p$.
- (k, l)-approval: $(a, \ldots, a, b, \ldots, b, 0 \ldots, 0)$, 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: $(b, b - 1, \ldots, 0)$, where if $b > m - 1$, b-Borda is reduced to the regular Borda rule with $(m - 1, m - 2, \ldots, 1, 0)$.

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 [32]

[32] 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

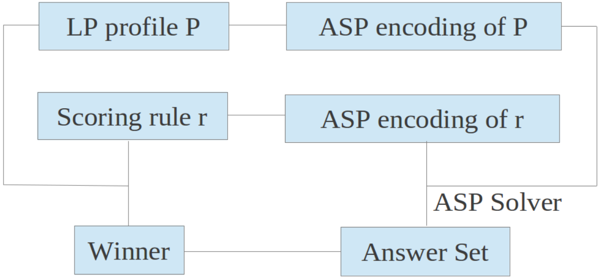


Figure : The winner problem

- Solvers: *clingo*[33], *clingcon*[34]

[33] M. Gebser et al. "Potassco: The Potsdam Answer Set Solving Collection". In: AI Communications (2011)

[34] Max Ostrowski and Torsten Schaub. "ASP modulo CSP: The clingcon system". In: TPLP (2012)

Modeling the Problems in W-MAXSAT

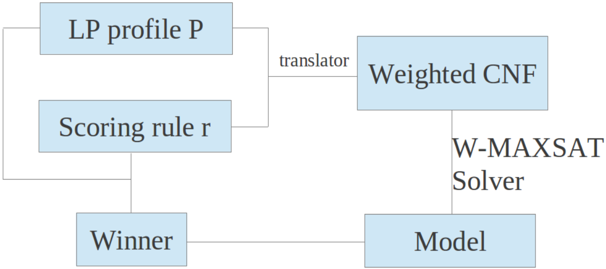


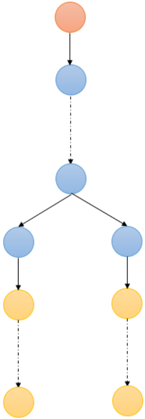
Figure : The winner problem

- Solver: *toulbar*[35]

[35] 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

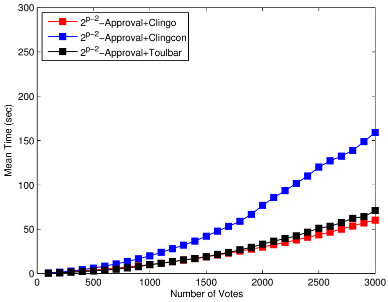


0_i	$0_r > 1_r$
1_i	$1_r > 0_r$

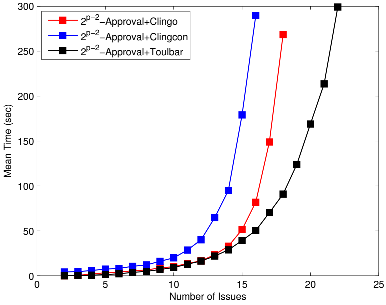
$0_j 0_k$	$0_s > 1_s$
$0_j 1_k$	$1_s > 0_s$
$1_j 0_k$	$0_s > 1_s$
$1_j 1_k$	$1_s > 0_s$

Figure : Random LP tree

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



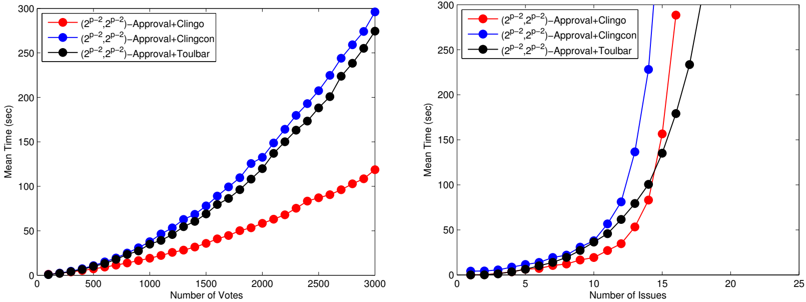
(a) Fixed #issues (10)



(b) Fixed #votes (1000)

Figure : Solving the winner problem

Varying p and n: $(2^{p-2}, 2^{p-2})$-approval [36]

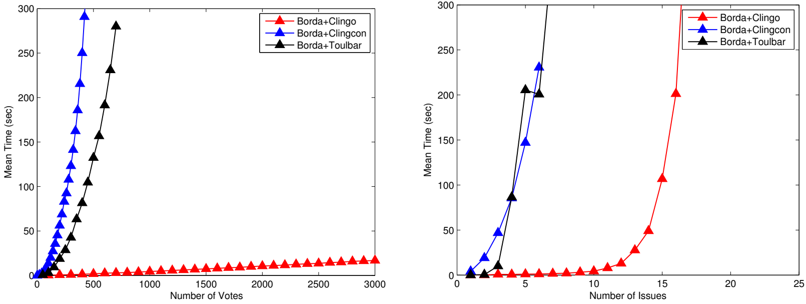


(a) Fixed #issues (10) (b) Fixed #votes (1000)

Figure : Solving the winner problem

36 scoring vector: $(2, \ldots, 2, 1, \ldots, 1, 0, \ldots, 0)$ 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

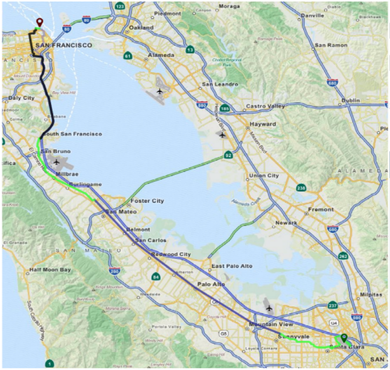
Personalization in Trip Planning

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*[37].
4. Available later for trip planning in the Bay Area, LA, and Denver.

[37] 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.



From:

San Jose Airport, San Jose, CA

To:

Pier 39, San Francisco, CA

Get directions

Route Summary
Start point: Terminal Drive
End point:
Driving time: 58 minutes
Distance: 48.6 miles

Segment	Mode	Time	Cost
1	bike	1 h 0 min 46 sec	\$0
2	transit	42 min	\$5.25
3	bike	56 min	\$0
4	uberx	15 min	\$23.92
5	bike	2 min	\$0
	Total:	2 h 57 min	\$29.17

Data-Driven Preference Engineering

1. Recommender Systems[38]:
   1. Collaborative
   2. Content-based
   3. Hybrid
2. Machine Learning:
   1. Supervised learning (e.g., decision trees, random forests)
   2. Label ranking[39]
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[40], TOPSIS-like models[41])

[38] 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)

[39] Eyke Hüllermeier et al. "Label ranking by learning pairwise preferences". In: Artificial Intelligence (2008)

[40] Agnes Leroy, Vincent Mousseau, and Marc Pirlot. "Learning the parameters of a multiple criteria sorting method". In: Algorithmic decision theory. 2011

[41] 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[42]:
    1. Voting
    2. Fair devision
    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

[42] Kenneth J Arrow, Amartya Sen, and Kotaro Suzumura. Handbook of Social Choice & Welfare. Vol. 1 & 2. Elsevier, 2010

Questions?

Thank you!