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

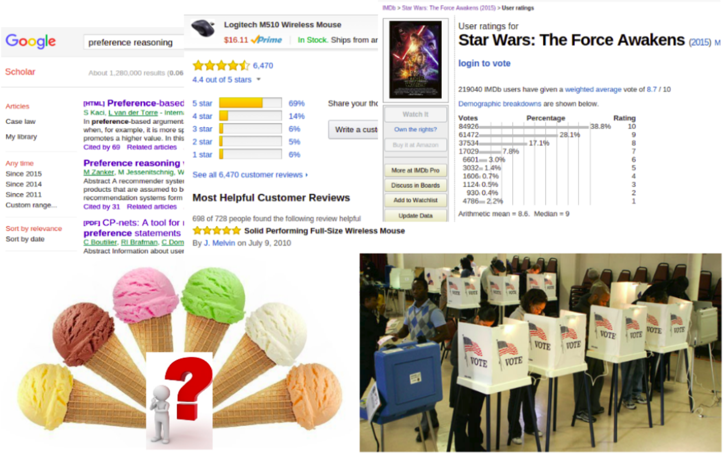


Figure : Preferences of different forms

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)

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,

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

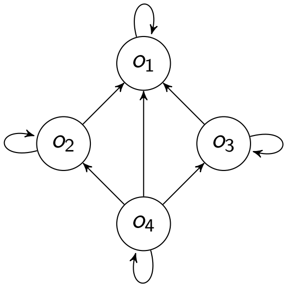
Properties of binary relations:

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 $o_1 = o_2$.

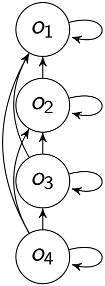
Relations and Orderings

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

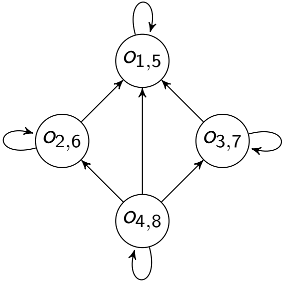


(b) total (pre)order

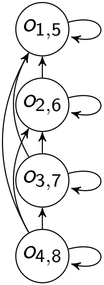
Relations and Orderings

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

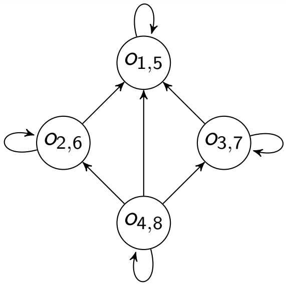


(b) total preorder

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

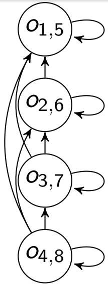
$$o_1 \preceq o_5,$$
$$o_4 \prec o_2,$$
$$o_4 \approx o_8,$$
$$o_6 \sim o_7.$$

(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

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}.

$\vdots$

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

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

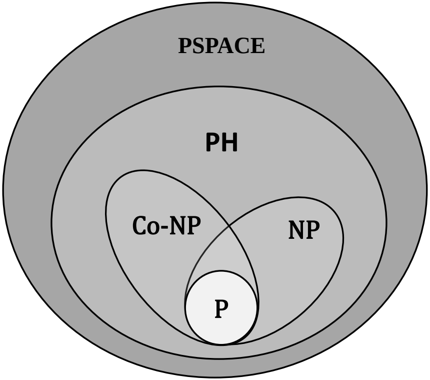


Figure : Computational complexity diagram

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] `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`, slightly adapted in the talk.

Qualitative Preferences

Individual:

Figure : Dominance Testing

Qualitative Preferences

Collective:

Figure : Social Choice and Welfare

Research on Preferences

Q: How do we represent preferences over combinatorial domains?

1. Quantitative:
    1. Utility/Cost Functions
    2. Possibilistic Logic[7]
    3. Fuzzy Preference Relations[15]
    4. Penalty Logic[17]
2. Qualitative:
    1. Preference Trees (e.g., LP-trees[2, 12], CLP-trees[5], PLP-trees[13], P-trees[8, 14])
    2. Conditional Preference Networks[3]
    3. Answer-Set Optimization Theories[6]
3. Hybrid:
    1. Preference Forests

Research on Preferences

Q: How do we learn/predict preferences over combinatorial domains?

1. Recommender Systems[1]:
    1. Collaborative
    2. Content-based
    3. Hybrid
2. Machine Learning:
    1. Supervised learning
    2. Label ranking
3. Preference Elicitation:
    1. Positive learning
    2. Human-in-the-loop
4. Preference Learning[9]:
    1. Conditional Preference Networks
    2. Preference Trees (e.g., LP-trees, CLP-trees, PLP-trees)

Research on Preferences

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

1. Preference Optimization:
   1. Consistency testing
   2. Dominance testing
   3. Optimality testing
2. Preference Aggregation[4]:
   1. Social choice and welfare
   2. Voting systems
3. Preference Misrepresentation[4]:
   1. Control
   2. Manipulation
   3. Bribery

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)).

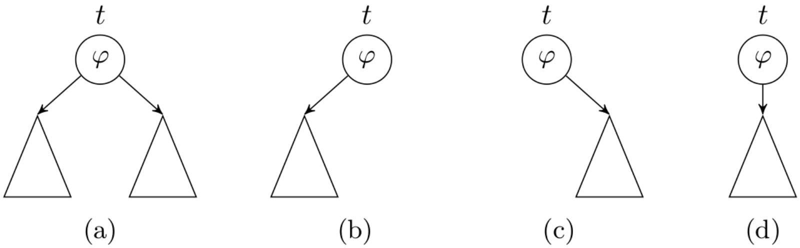


Figure : Compact P-trees

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**[2]	$\boldsymbol{\Delta_2^P(P^{NP})}$**-c**[3]
ASO-theory	P	coNP-c	$\Sigma_2^P(NP^{NP})$-c

Figure : Computational complexity results

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

[3] 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)

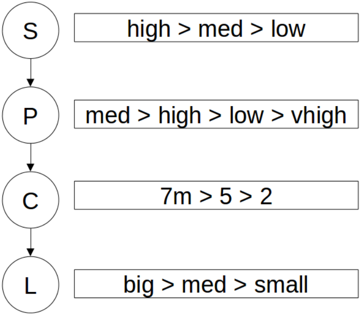


Figure : A UIUP PLP-tree

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

Partial Lexicographic Preference Trees (PLP-Tree)

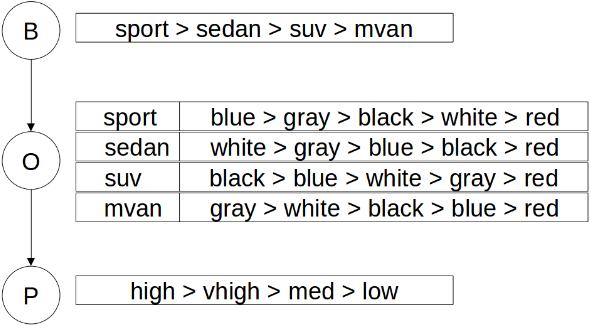


Figure : A UICP PLP-tree

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

Partial Lexicographic Preference Trees (PLP-Tree)

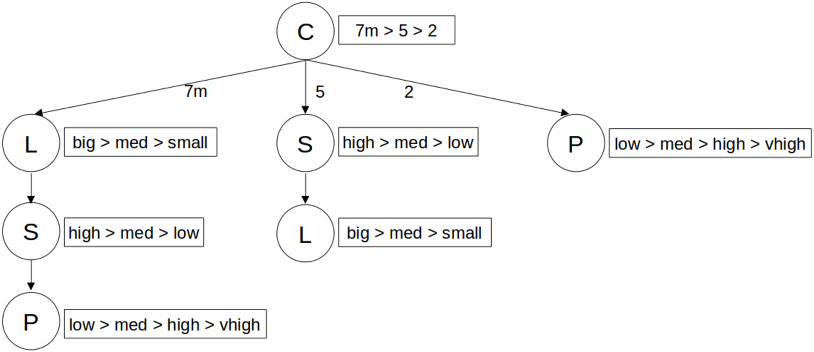


Figure : A CIUP PLP-tree

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

Partial Lexicographic Preference Trees (PLP-Tree)

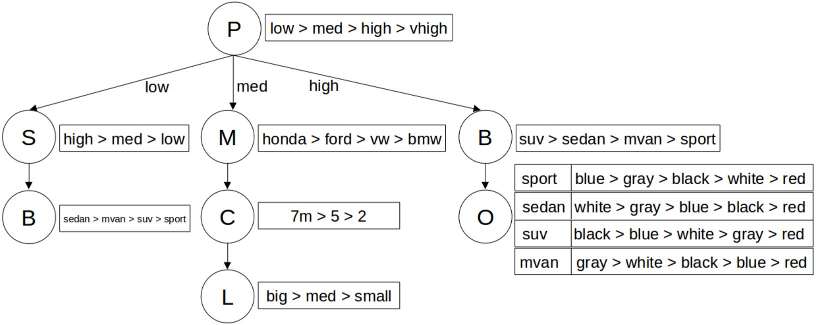


Figure : A CICP PLP-tree

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

Complexity Results

	UP	CP
UI	P	P
CI	NPC[4]	P

(a) ConsLearn

	UP	CP
UI	NPC	NPC
CI	NPC	NPC

(b) SmallLearn

	UP	CP
UI	NPC[5]	NPC
CI	NPC	NPC

(c) MaxLearn

Figure : Complexity results for passive learning problems

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

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

Experimental Results on Trees

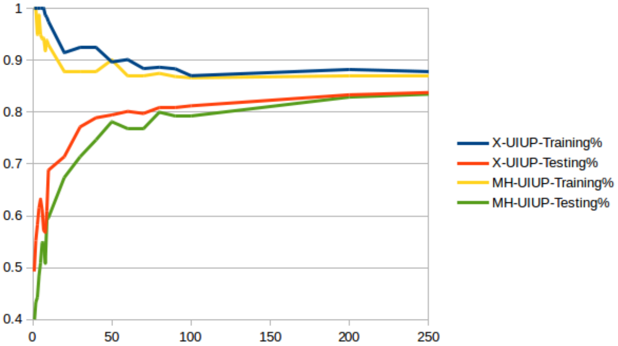


Figure : Learning curve for UIUP using ASP and greedy heuristic

Experimental Results on Trees

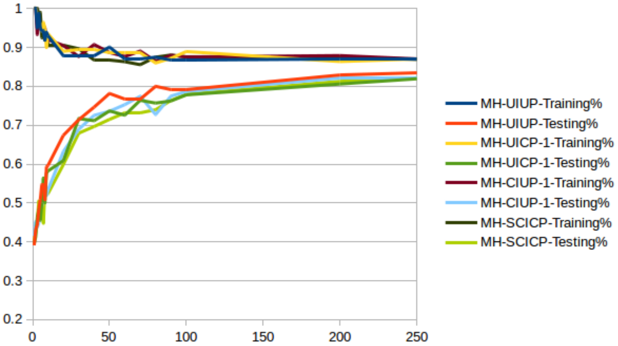


Figure : Learning curve for all four classes using greedy heuristic

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

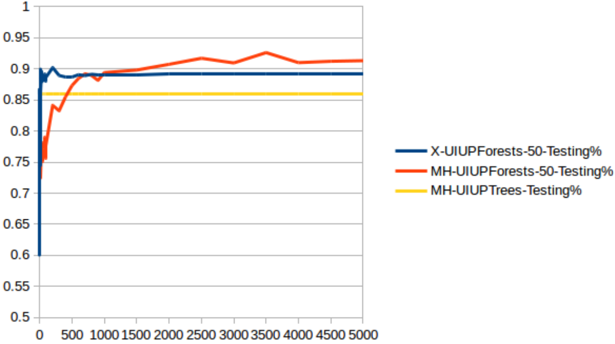


Figure : Learning UIUP using ASP and greedy heuristic

Experimental Results on Forests

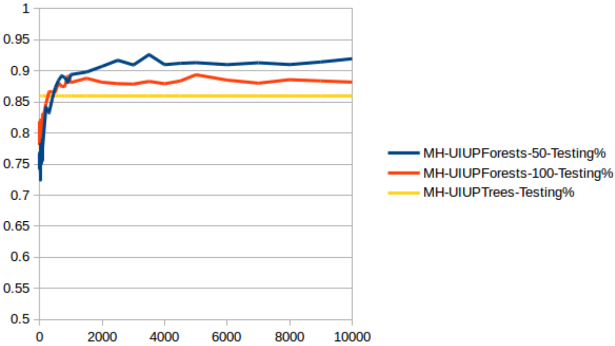


Figure : Learning all four classes using greedy heuristic

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 [6]

[6] 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

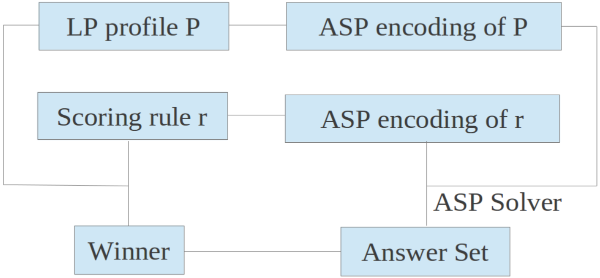


Figure : The winner problem

- Solvers: *clingo*[7], *clingcon*[8]

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

[8] Max Ostrowski and Torsten Schaub. "ASP modulo CSP: The clingcon system". In: TPLP 12.4-5 (2012), pp. 485–503.

Modeling the Problems in W-MAXSAT

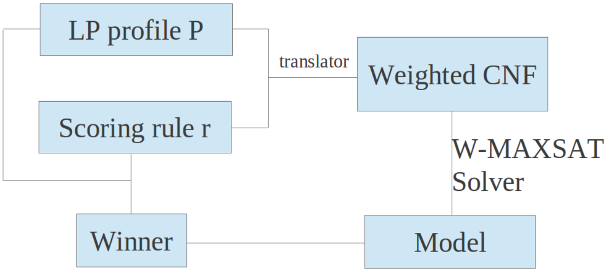


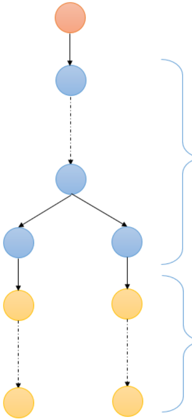
Figure : The winner problem

- Solver: *toulbar*[9]

[9] 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

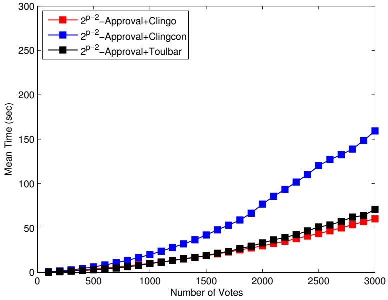


0_i	$0_r > 1_r$
1_i	$1_r > 0_r$

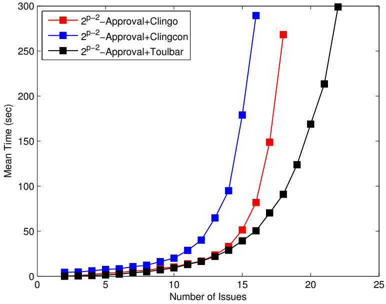
$0_j 0_k$	$0_s > 1_s$
$0_j 1_k$	$1_s > 0_s$
$1_j 0_k$	$0_s > 1_s$
$1_j 1_k$	$1_s > 0_s$

Figure : Random LP tree

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



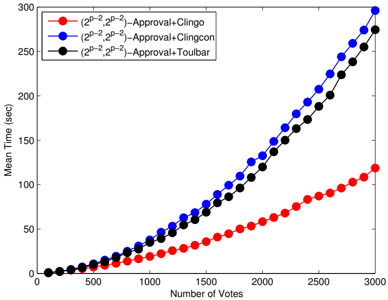
(a) Fixed #issues (10)



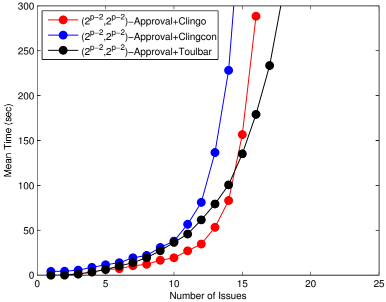
(b) Fixed #votes (1000)

Figure : Solving the winner problem

Varying p and n: $(2^{p-2}, 2^{p-2})$-approval [10]



(a) Fixed #issues (10)

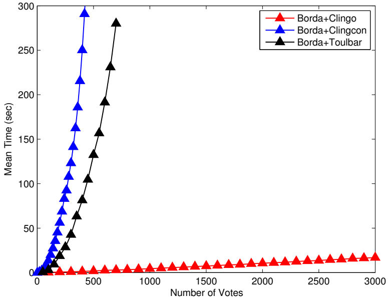


(b) Fixed #votes (1000)

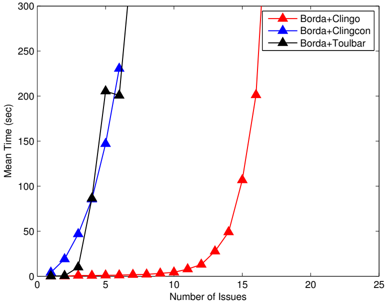
Figure : Solving the winner problem

[10] 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

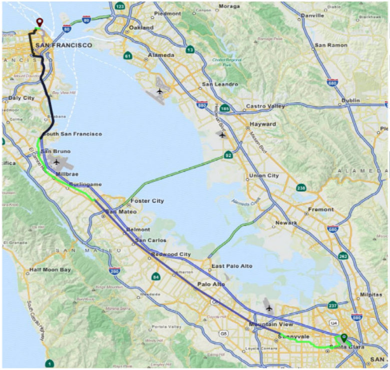
Personalization in Trip Planning

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*[11].
4. Available later for trip planning in the Bay Area, LA, and Denver.

[11] Xudong Liu. "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

Preference Learning

Goal: preference predicting and feasibility of various preference models.

1. Recommender Systems:
   1. Collaborative
   2. Content-based
   3. Hybrid
2. Machine Learning:
   1. Supervised learning
   2. Label ranking
3. Preference Elicitation:
   1. Positive learning
   2. Human-in-the-loop
4. Preference Learning:
   1. Conditional Preference Networks
   2. Preference Trees (e.g., LP-trees, CLP-trees, PLP-trees)
   3. Stochastic Models (e.g., Choquet integral, TOPSIS-like models)

Preference Reasoning

Goal: personalized optimization and collaborative decision making.

1. Preference Optimization:
    1. Consistency testing
    2. Dominance testing
    3. Optimality testing
2. Preference Aggregation:
    1. Social choice and welfare
    2. Voting systems
3. Preference Misrepresentation:
    1. Control
    2. Manipulation
    3. Bribery

Preference Applications

Goal: exploring possibilities of collaboration with experts in other areas.

1. Automated Planning and Scheduling:
    1. Travel scheduling
    2. Manufacturing
    3. Traffic control
2. Computer Vision and Image Processing:
    1. Image retrieval
    2. Image and video understanding

Questions?

Thank you!