Logic-Based Preference Languages with Intermediate Complexity

Joel Uckelman
ILLC, University of Amsterdam
Plantage Muidergracht 24
1018TV Amsterdam, Netherlands

Andreas Witzel
ILLC, University of Amsterdam
Plantage Muidergracht 24
1018TV Amsterdam, Netherlands
and CWI, Kruislaan 413,
1098SJ Amsterdam, Netherlands

Abstract

Logic-based preference representation languages are used to represent utility functions in combinatorial auctions, and can also be used for voting. A canonical problem in this context is, given a set of propositional formulas with associated weights, to find an assignment[1] that maximizes the sum of those weights which are associated to satisfied formulas. The general case is intractable, and natural restrictions of the languages tend either to leave the complexity unchanged or to reduce it to triviality. After proposing a revision of the decision problem considered in existing research, we use a new approach to find languages with P-complete maximization problem.

Introduction

The MAX-UTIL problem for preference representation languages based on weighted logic formulas consists in finding an assignment which maximizes the sum of those weights which are associated to satisfied formulas. The complexity of a decision problem version of MAX-UTIL is considered in (Chevaleyre, Endriss, & Lang 2006; Uckelman & Endriss 2007). The picture which emerges is bipolar: Every language for which a positive result is presented has either a trivial decision problem or an NP-complete one. This naturally led us to wonder whether there are any preference representation languages which occupy the (previously unexplored) middle ground. Further consideration has led us to formulate a modified version of the decision problem—one which we feel better captures the interest of the problem. Here we present that modified decision problem and apply a new approach to find two languages for which it is P-complete.

Who (if anyone) needs to solve MAX-UTIL depends on the context in which these preference representation languages are being applied. Take auctions, for example: Whether MAX-UTIL needs to be solved by the center (e.g., the auctioneer) immediately in order to determine the winner depends on his concrete algorithm; the center *does* solve MAX-UTIL if the resources are shareable. Specifically, MAX-UTIL *is* the Winner Determination Problem for combinatorial auctions where the auctioneer has free disposal, the bidders do not have free disposal, and allocated goods are shared among all bidders. This might at first sound like a strange sort of auction, one where all bidders receive every good won by any bidder—but this is precisely what an election is. The candidates are the goods, and everyone shares whatever good (or goods, in the case of a multi-winner election) is allocated. Solving MAX-UTIL over the admissible models, i.e., the ones which elect the correct number of candidates, tells you who has won the election. Many popular voting methods have analogues in this framework.

Even in cases where it is not necessary to solve MAX-UTIL in order to solve the Winner Determination Problem, the complexity of MAX-UTIL provides a lower bound on how complex the Winner Determination Problem is: Observe that in the (degenerate) single-bidder case, the two problems coincide. If only one bidder shows up to the auction, then determining which items she wins is precisely the same as finding her optimal state. Therefore, the Winner Determination Problem can never be easier than MAX-UTIL, as it contains MAX-UTIL as a sub-problem.

Finally, for an agent herself it is useful to solve MAX-UTIL if she builds her bids not directly from an explicitly represented utility function, but instead from constraints or through elicitation. In that case, the agent may only be able to figure out her optimal state by solving MAX-UTIL. Here, all value is measured along a single axis, utility. Were we to consider an extension of weighted formulas to encompass multiple, incommensurable measures, as in multi-criterion decision making, it would be even less likely that an agent would be aware of her optimal states, and hence solving MAX-UTIL becomes even more important in that setting.

Preliminaries

The importance of finding appropriate bidding languages for combinatorial auctions is well-known (Nisan 2006). We take a logic-based approach:

Definition 1 (Atoms, Weighted Formulas, Goal Bases and Utility Functions)**.**

- *By $\mathcal{PS}$ we denote the finite set of propositional symbols (*atoms*), which in our context represent the goods on the auction block.*

[1] Existing research uses the term *model*; we want to differentiate from the slightly different notion of a *model of a logic program*.

- *For $M \subseteq \mathcal{PS}$ and a propositional formula φ of a language $\mathcal{L}_{\mathcal{PS}}$ over $\mathcal{PS}$ (using $\wedge$, $\vee$, and $\neg$ as connectives), we write $M \models \varphi$ to say that φ is satisfied by assigning true to all atoms in M and false to all others.*
- *A* weighted formula *is a pair (φ, w) where $\varphi \in \mathcal{L}_{\mathcal{PS}}$ and $w \in \mathbb{R}$.*
- *A* goal base *is a set $G = \{(\varphi_i, w_i)\}_i$ of weighted satisfiable formulas.*
- *The* utility function u_G *generated by the goal base G is, for $M \subseteq \mathcal{PS}$,*

$$u_G(M) = \sum \{w_i : (\varphi_i, w_i) \in G \text{ and } M \models \varphi_i\} \ .$$

Note that the fact that goal bases contain only satisfiable formulas is important. Without it, any computational task involving goal bases which involves finding models would immediately become NP-hard; with this constraint, we can avoid solving SAT for the formulas in our goal bases. This is a reasonable limitation, as it would be strange for users of a bidding language to supply unsatisfiable formulas in their bids.

Definition 2.

- *The* MAX-UTIL function problem *for a class C of goal bases is as follows: Given $G \in C$, find an assignment $M \subseteq \mathcal{PS}$ such that $u_G(M)$ is maximized.*
- *The* decision problem *used in previous research is as follows: Given $G \in C$ and a number K, is $u_G(M) \geq K$ for a maximizing assignment M?*

In addition to these concepts from combinatorial auctions, we need the following notions and results from propositional logic programming (PLP), taken from (Dantsin *et al.* 2001).

Definition 3 (Horn Clauses, Least Models)**.** *A* strict (resp. general) Horn clause *is a non-empty disjunction of exactly (resp. at most) one atom and zero or more negated atoms.*

For a set S of strict Horn clauses, a least model $LM(S)$ *of S is a smallest set $M \subseteq \mathcal{PS}$ such that $M \models S$, that is, $M \models \varphi$ for all $\varphi \in S$.*

Fact 4. *Any set S of strict Horn clauses has a unique least model.*

Definition 5. *The* PLP decision problem *is as follows: Given a set S of strict Horn clauses and some $p \in \mathcal{PS}$, is $p \in LM(S)$?*

Fact 6. *The* PLP *decision problem is P-complete.*

Finally, we will use the following decision problem along with its complexity result, to be found, e.g., in (Greenlaw, Hoover, & Ruzzo 1992).

Definition 7. *The* HORNSAT decision problem *is as follows: Given a set S of general Horn clauses, is S satisfiable?*

Fact 8. *The* HORNSAT *decision problem is P-complete.*

Logically speaking, Horn clauses express facts and dependencies in the following ways:

- Strict Horn clauses with no negated atoms, i.e., consisting only of one atom, represent plain *facts*. In the context of auctions, these are statements about single goods, in voting, single candidates: "I'll pay \$50 for the Elvis statue.", "I cast a vote for Obama."
- Strict Horn clauses containing negated atoms correspond to *implications*. In our context they can be viewed as statements conditioned on several goods with one good as consequence: "If you don't eat your meat, you can't have any pudding." Additionally, strict horn clauses with negated atoms lend themselves to describing situations in which both *goods* and *bads* must be divided: "For \$1, either I get the last piece of cake, or I don't have to clean the bathroom."
- Non-strict Horn clauses, i.e., disjunctions containing only negated atoms, correspond to *negated conjunctions*; we can think of them as "negative synergies", or exclusions of certain combinations of goods: "A committee with *both* Alice *and* Bob on it would be a disaster.", "I would appreciate not having *both* my defense *and* my job interview today.", "If I have to change planes in London, it's worth \$50 to me to avoid doing it at Heathrow.",

These ways of interpreting Horn clauses have proved their usefulness in the area of logic programming. We believe that they also make them a versatile and powerful base for preference representation languages.

Revising the MAX-UTIL Decision Problem

In our view, the decision problem from Definition 2 does not adequately capture the function problem defined there.

If a decision problem is used to simplify the formulation of some function problem to a mere yes/no question, then the complexity of finding a solution should be preserved, and these problems should be related in the sense that solving one enables one to solve the other one easily (Papadimitriou 1993).

Currently, the two formulations of MAX-UTIL are not related in this sense.

With the classes considered in previous research, the decision problem did capture the complexity of the function problem and could indeed be solved most expediently by solving the function problem; however, consider the following class C of goal bases:

$$C = \left\{\{(\varphi_i, w_i)\}_i : \bigwedge \varphi_i \text{ is satisfiable and all } w_i \geq 0\right\} \ .$$

The decision problem here is trivial (sum the weights and check whether the sum exceeds the given K) and gives no guidance as to the solution of the function problem, nor does it reflect its complexity.

We therefore propose the following decision problem:

Definition 9 (MAX-UTIL Decision Problem)**.** *Given a goal base G and an atom $p \in \mathcal{PS}$, is p true under the maximizing assignment (fix an arbitrary one if not unique)?*

To see why it is necessary to fix one maximizing assignment in case there are several, consider the goal base $\{(p \wedge \neg q, 1), (\neg p \wedge q, 1)\}$) and note that both p and q are true under *some* maximizing assignment, but both taken together do *not* maximize the utility.

Note. It may seem ugly to fix an arbitrary assignment, and indeed one could, for example, require the least assignment with respect to some ordering; however by doing so the complexity of the problem may actually increase. This becomes evident with the PLP goal base class presented in the following; see the discussion at the end of this paper.

By executing a solution procedure for this decision problem $|\mathcal{PS}|$ times, one can construct a solution to the original function problem; vice versa, solving the function problem obviously enables one to solve the decision problem. Hence, the revised decision problem given in Definition 9 is related to the function problem in the sense described above.

In terms of computational complexity, we can say that the decision problem is, for the most general language in which all formulas and all weights are allowed, NP-complete; and that the corresponding function problem is in TFNP, which is the class of function problems on polytime-decidable predicates for which there is guaranteed to be a witness. (For a discussion of complexity classes associated with function problems, see (Megiddo & Papadimitriou 1991).)

Finding P-complete Goal Base Classes

The MAX-UTIL problem is NP-complete for the general class of goal bases allowing arbitrary propositional formulas and weights. (This can be seen via a simple reduction from MAXSAT to MAX-UTIL, in which every formula in the input is given weight 1.) Attempts to find tractable subclasses in previous research consisted in putting natural restrictions on the formulas and weights, e.g., allowing only conjunctions of (negated) atoms and positive weights. As mentioned above, the resulting classes were either still intractable or trivial. In order to make preference representation languages tractable while retaining an interesting measure of expressive power, it seems necessary to find classes with some intermediate complexity.

We thus propose the following approach: Instead of putting restrictions on the goal bases and then examining the complexity, we take a problem which has a certain complexity and find a class of goal bases whose MAX-UTIL problem corresponds to it.

Intuitively, it is evident that Horn clauses are more versatile and expressive than the above-mentioned natural restrictions. For example, $(\neg a \vee b, 1)$ translates into positive cubes (conjunctions) as $\{(\top, 1), (a, -1), (a \wedge b, 1)\}$, while $(\neg a \vee \neg b, 1)$ becomes $\{(\top, 1), (a \wedge b, -1)\}$. While these are not cumbersome on their own, it can become so when several Horn clauses are translated together, since in translation the weight of each Horn clause is distributed over multiple positive cubes. Translation into another simple language, positive clauses with positive weights, will not typically be possible, as general Horn clauses are not monotone formulas and so require a language which offers either negation as a connective or permits negative weights.

Furthermore, there are various P-completeness results involving Horn clauses, two of which we stated above.

For these reasons, in the following we will apply our approach to find two P-complete goal base classes related to Horn clauses.

PLP Goal Bases

Definition 10. *The class $\mathcal{G}_{\mathrm{PLP}}$ of* PLP goal bases *consists of all goal bases*

$$G = \{(\varphi_i, w_i)\}_i \cup \left\{\left(p, -\tfrac{m}{|\mathcal{PS}|+1}\right)\right\}_{p \in \mathcal{PS}},$$

where

- *the φ_i are strict Horn clauses,*
- *the w_i are positive, and*
- *m equals* $\min_i\{w_i\}$.

$LP(G) := \{\varphi_i\}_i$ *is the* underlying logic program *consisting of all positively weighted formulas. The remaining terms are* penalty terms.

The penalty terms are needed for technical reasons, and we will return to them in the discussion.

Fact 11. *The weights of the penalty terms sum up to an absolute value less than any of the w_i. That is, for all i,*

$$w_i > \sum_{p \in \mathcal{PS}} \frac{m}{|\mathcal{PS}|+1}.$$

Corollary 12. *The (unique) maximizing valuation of any $G \in \mathcal{G}_{\mathrm{PLP}}$ is the least model of the underlying logic program, i.e., $LM(LP(G))$.*

Proof. $LM(LP(G))$ obviously satisfies all formulas of G that have positive weights. Since it is a *least* model, due to Fact 11, none of its subsets get a higher value; due to the penalty terms, none of its supersets get a higher value; and due to Fact 4, it is unique. □

Lemma 13. *The* MAX-UTIL *decision problem for PLP goal bases is in P.*

Proof. Given $G \in \mathcal{G}_{\mathrm{PLP}}$ and $p \in \mathcal{PS}$, $LP(G)$ can be computed in linear time, and then $p \in LM(LP(G))$ is decidable in polynomial time due to Fact 6. Due to Corollary 12, this yields the answer to the MAX-UTIL decision problem. □

Lemma 14. *PLP can be reduced in logarithmic space to the* MAX-UTIL *decision problem for PLP goal bases.*

Proof. Given a logic program $S = \{\varphi_i\}_i$ and $p \in \mathcal{PS}$, let

$$G := \bigcup_{i=1}^{n}\{(\varphi_i, 1)\} \cup \bigcup_{p \in \mathcal{PS}} \left\{\left(p, -\tfrac{1}{|\mathcal{PS}|+1}\right)\right\}.$$

Obviously, $G \in \mathcal{G}_{\mathrm{PLP}}$, and due to Corollary 12, solving the MAX-UTIL decision problem instance (G, p) yields the answer to the PLP decision problem instance (S, p). □

Corollary 15. *The* MAX-UTIL *decision problem for PLP goal bases is P-complete.*

Proof. Follows immediately from Lemmas 13 and 14. □

HS Goal Bases

Definition 16. *The class* $\mathcal{G}_{\mathrm{HS}}$ *of* HORNSAT goal bases *consists of all sets* G *of weighted general Horn clauses with positive weights, subject to the following condition:*

Let w_i *denote the weights of the strict Horn clauses in* G *and* w'_j *denote the remaining weights. Then we require that*

$$\sum_j w'_j < \min_i \{w_i\} \ .$$

That is, the sum of weights of non-strict clauses (i.e., those containing no positive atom) is less than the weight associated to any strict clause.

This condition does not appear to be very intuitive, and we will return to it in the discussion. For the time being, note that it is only needed to ensure that the complexity stays within P (Lemma 17); it may be possible to find a more intuitive condition to this effect.

Lemma 17. *The* MAX-UTIL *decision problem for HS goal bases is in P.*

Proof. Given $G \in \mathcal{G}_{\mathrm{HS}}$, use e.g., unit propagation to find a satisfying assignment if one exists. If it does exist, this is the maximizing assignment since all weights are positive. If it does not exist, let $G' \subset G$ be the subset containing all *strict* Horn clauses. Due to the condition in Definition 16, $LM(G')$ is a maximizing assignment for G, since

- it satisfies all strict Horn clauses, and
- among all such assignments, it satisfies the most non-strict Horn clauses.

The second item holds due to the fact that we have a *least* model of G', that is, one that satisfies the *greatest* set of negated atoms, and non-strict Horn clauses are just disjunctions of those. □

Lemma 18. HORNSAT *can be reduced in logarithmic space to* MAX-UTIL *for HS goal bases.*

Proof. Given a set $S = \{\varphi_1, \ldots, \varphi_n, \varphi'_1, \ldots, \varphi'_m\}$ of strict (φ_i) and non-strict (φ'_i) Horn clauses, build the HS goal base

$$G := \bigcup_{i=1}^{n} \{(\varphi_i, 1)\} \cup \bigcup_{i=1}^{m} \{\left(\varphi'_i, \tfrac{1}{m+1}\right)\} ,$$

obtain the maximizing assignment by solving MAX-UTIL for G and each $p \in \mathcal{PS}$, and check whether it satisfies all formulas in G. Since the assignment is maximizing and all weights are positive, it will do so iff G is satisfiable. □

Corollary 19. *The* MAX-UTIL *decision problem for HS goal bases is P-complete.*

Proof. Follows from Lemmas 17 and 18. □

Discussion

As mentioned earlier, we believe that Horn clauses form a versatile and powerful base for preference representation languages, since their form is restricted in a clear way, but they retain the ability to express natural forms of dependency. The existence of various P-completeness results involving Horn clauses suggests that they lend themselves to our approach. We therefore focused on these, without meaning to suggest that other classes of formulas might not be worth considering. There are certainly other P-complete fragments of the full weighted formula language which are induced by other P-complete problems and embody different kinds of synergies than those examined here. We thus hope that our approach can be fruitfully applied to obtain further results.

While some of our examples focused on auctions, Horn clauses also have useful interpretations in multi-winner voting. They can express dependencies among candidates, e.g., to say that Alice should be on a committee whenever Bob is, or that Alice should *not* be on a committee if Bob is.

The goal base classes we presented may at first glance seem artificial and unnatural, and they may then simply be viewed as proof of concept for our approach, and proof of existence for logic-based preference representation languages of intermediate complexity.

However, the penalty terms which occur in PLP do reflect an intuitively justifiable desideratum, since they make, ceteris paribus, assigning fewer items favorable. If no-one benefits from obtaining some additional item, why should the auctioneer give that item away for nothing instead of keeping it for some later auction? In that sense, it might even be desirable to require a *least* maximizing assignment in the definition of the MAX-UTIL problem itself. With such an alternative definition, one could remove the penalty terms from PLP goal bases and obtain a quite natural P-complete goal base class. This also shows that, as noted under Definition 9, requiring the *least* (instead of an arbitrary) maximizing assignment has an effect on the complexity of MAX-UTIL: With such a requirement, it would be P-complete for PLP goal bases without penalty terms, while as it stands, it is trivially solved by making all atoms true.

As for HS goal bases, as mentioned above, the unintuitive condition in Definition 16 is only used to prove Lemma 17, and it may be possible to find a more intuitive condition to that effect. However, this condition might even be acceptable if bids or preferences can be described "lexicographically" on two levels: Strict Horn clauses (facts and implications) describe the primary bid in form of a logic program. Then, non-strict Horn clauses (exclusions of certain combinations) can be added for fine-tuning and favoring certain models of the logic program over others. Note that this secondary bid matters, since HS goal bases, contrarily to PLP goal bases, do not enforce *least* models.

In future research, we plan to look closer at issues such as representational power, pursue our ideas for making our goal base classes more intuitive, and use our approach in order to find other preference representation languages with intermediate complexity.

Acknowledgements

We thank our advisor Ulle Endriss for helpful discussions, and three anonymous referees for their constructive comments.

This research was supported by a GLoRiClass fellowship funded by the European Commission (Early Stage Research Training Mono-Host Fellowship MEST-CT-2005-020841).

References

Chevaleyre, Y.; Endriss, U.; and Lang, J. 2006. Expressive power of weighted propositional formulas for cardinal preference modelling. In *Proc. 10th Intl. Conference on Principles of Knowledge Representation and Reasoning (KR-2006)*, 145–152. AAAI Press.

Dantsin, E.; Eiter, T.; Gottlob, G.; and Voronkov, A. 2001. Complexity and expressive power of logic programming. *ACM Computing Surveys* 33:374–425.

Greenlaw, R.; Hoover, H. J.; and Ruzzo, W. L. 1992. A compendium of problems complete for P. Technical report, University of Alberta, Computer Science Department.

Megiddo, N., and Papadimitriou, C. H. 1991. On total functions, existence theorems, and computational complexity. *Theoretical Computer Science* 81(2):317–324.

Nisan, N. 2006. Bidding languages for combinatorial auctions. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press. 215–232.

Papadimitriou, C. H. 1993. *Computational Complexity*. Addison Wesley.

Uckelman, J., and Endriss, U. 2007. Preference representation with weighted goals: Expressivity, succinctness, complexity. In *Preference Handling for Artificial Intelligence. Papers from the 2007 AAAI Workshop*, 85–92.