Preference Reasoning Under Partial Alternatives

Michael Huelsman
100 Saint Anselm Drive
Goffstown, NH 03102

Abstract

Selecting a single alternative among many is a key cognitive task. Many formal approaches for solving this problem have been explored, but these approaches consider having, or eliciting, complete preference information. As a natural extension, work has also been done to consider situations where complete preference information is unknown. What has not been studied is the effect of incomplete information wrt the alternatives an agent may select from. This work focuses on the computational problems that arise when preference information is known, but alternatives are only partially specified, such as when one is searching online classified ads. We both define these problems and specify some general computational complexity results. While the complexity of the defined problems are not tightly bound, we do provide a case study which demonstrates how partial alternatives affect different preference representations differently.

Introduction

Rational agents must be capable of making decisions. Often these decisions are based, not only on the agent's needs, but their preferences. The importance of modeling an agent's preferences has led to a great deal of research across many fields such as operations research, economics, psychology, and artificial intelligence(Domshlak et al. 2011). A common assumption in this research is that a particular agent's preferences are either known, or can be elicited, i.e. we have full information. Of course, there has also been research into partial knowledge of an agent's preferences(Fagin et al. 2006; Cullinan, Hsiao, and Polett 2014). These representations of partial preference knowledge tend to involve specialized preference representations(Liu and Truszczynski 2015) including through probabilistic reasoning(Cornelio et al. 2021).

This work also deals with the domain of preference reasoning under limited information or uncertainty, but instead of the agent's preferences being unknown, we consider a domain where we have limited knowledge about the alternatives under consideration. This type of limited information occurs often in real-world scenarios. For example, consider the purchase of a used camera from an online auction site. The availability of product information is likely to be extremely variable between different sellers. I.e some information will be present in certain listings that is not available in others, and vice versa. In this setting you do not have complete information about each alternative. In these cases gathering further information may be impossible, however agent's may still need to make the best decision possible. In particular, we concern ourselves with problems where we know an agent's preferences under complete information alternatives and use that to reason about incomplete information alternatives, herein referred to as incomplete alternatives.

This exercise leads us to two problems, possible dominance and necessary dominance. Similar to their definitions in other domains of incomplete preference information, possible dominance occurs when one alternative has the potential to dominate (or be preferred to) another alternative, while necessary dominance occurs when one alternative will always dominate another alternative. These problems focus solely on the comparison of pairs of alternatives. Problems which look at the entire domain of alternatives, such as optimality tend to remain the same under incomplete alternatives as the domain of all complete alternatives is included in the domain of all incomplete alternatives.

This work's particular contribution is to define some general complexity results which arise from the introduction of incomplete alternatives. These results do not allow us to give strict complexity bounds to all instances of dominance under incomplete alternatives, but do significantly limit the scope of possibility. As an example of the effect of incomplete alternatives on computational complexity, we provided two case studies where we determine strict complexity bounds for the introduced problems.

This work is organized as follows. The next section provides required background information and definitions for the works. The third section presents our results, along with proofs. Finally, the fourth section contains a discussion of the results as well as our final conclusions and remaining open questions.

Background

Given the variability of human preferences there have been many models proposed for how to represent preferences(Kaci 2011) which is beyond the purview of this paper.

One common thread through all these representations is the use of preorders. A *preorder* is any binary relation $\succeq$ which is reflexive and transitive. In particular we define $\alpha \succeq \beta$, given two alternatives α and β, if α is at least as preferred as β. Moreover, a *total order* is any preorder where the relation is irreflexive and exists between all pairs of alternatives. Given a preference relation $\succeq$ we can define the related relation of dominance.

Definition 1 (Dominance (DOM))**.** *Given a preference relation $\succeq$ and two partial alternatives α and β, α dominates β if $\alpha \succeq \beta$ and $\beta \not\succeq \alpha$.*

Preference relations occur over a domain of alternative Ω, this work is only concerned with domains of alternatives that are combinatorial in nature. A *combinatorial domain* is defined by a finite, ordered set of attributes $(A_1, A_2, A_3, \ldots, A_n)$ where each attribute has a finite domain of possible values such that $D(A_i) = \{v_1, v_2, \ldots, v_m\}$. An alternative α of a combinatorial domain is a member of the set defined by the Cartesian product of the individual attribute domains, i.e. $\alpha \in D(A_1) \times D(A_2) \times \ldots \times D_(A_n)$.

We define *incomplete alternatives* within a given combinatorial domain Ω to be alternatives where some attribute values are unknown. There are any number of reasons why an attribute, or attributes, may be unknown, but an agent still needs to select between two alternatives. To this end, we define two problems, *possible incomplete dominance* and *necessary incomplete dominance*, below.

Definition 2 (Possible Incomplete Dominance (P-DOM))**.** *Given a preference relation $\succeq$ and two partial alternatives α and β, α possibly dominates β if there exists some completion of α and β such that $\alpha \succeq \beta$ and $\beta \not\succeq \alpha$.*

Definition 3 (Necessary Incomplete Dominance (N-DOM))**.** *Given a preference relation $\succeq$ and two partial alternatives α and β, α necessarily dominates β if for all completions of α and β, $\alpha \succeq \beta$ and $\beta \not\succeq \alpha$.*

For simplicity we will refer to these as P-DOM and N-DOM for the rest of this work. It is important to state that while these definitions are similar to possible and necessary dominance used elsewhere in the preference reasoning literature, those definitions deal with partially specified preference relations and thus are distinctly different problems from those defined above.

Later in this work we will provide some generic complexity results, wrt P-DOM and N-DOM, however these results are not strict and so we provide a case study of P-DOM and N-DOM complexity results for two specific preference representations: Lexicographic Preference Models (Fishburn 1974) and Ranked Preference Formulas (Huelsman and Truszczynski 2022). A *lexicographic preference model* (LPM) represents preferences over a combinatorial domain by assigning an importance order to the attributes and a preference order for each attribute domain.

Formally, an LPM is a pair $\pi = (r, \succ)$, where r is a strict ranking (a one-to-one mapping of attributes to integers) of attributes and $\succ = \{\succ_{v_1}, \succ_{v_2}, \ldots, \succ_{v_n}\}$ is a collection of total orders, with one order for each attribute's domain. Preference is formally defined over an LMP $\pi = (r, \succ)$ as $\alpha \succ_\pi \beta$ if there exists some attribute a such that $\alpha[a] \succ_a \beta[a]$ and for all attributes b such that $r(b) < r(a)$, $\alpha[b] = \beta[b]$.

Consider the domain defined over the power set of $O = \{a, b, c, d\}$. Suppose we have an LPM $\pi = (r, \succ)$ where the ranking r induces the order $d \succ c \succ b \succ a$ on attributes and we prefer the inclusion of an object over its exclusion. In this case the alternatives $\{a, b, d\}$ and $\{a, c\}$ are ordered such that $\{a, b, d\} \succ \{a, c\}$. They are ordered this way because the most important attribute where the two alternatives differ is d, and the inclusion of d is preferred over its exclusion, with no consideration paid to other attributes like c or b.

Ranking preference formulas (RPFs) are defined over a binary combinatorial domain $\mathcal{C}(\mathcal{V})$ (a combinatorial domain where all attributes have only possible two values) and consists of a vector of preference formulas $\boldsymbol{\varphi} = (\varphi_1, \varphi_2, \ldots, \varphi_k)$, where $\varphi_i \in \mathcal{L}(\mathcal{V})$, for $i = 1, \ldots, k$. Where $\mathcal{L}(\mathcal{V})$ is the language of propositional logic defined over the set $\mathcal{V}$ of attributes of the domain $\mathcal{C}(\mathcal{V})$. We define the *rank* of an alternative α, $r(\alpha)$ by setting $r(\alpha) = \min\{i : \alpha \models \varphi_i\}$. Given two alternatives $\alpha, \beta \in \mathcal{C}(\mathcal{V})$, we define $\alpha \succeq_{\boldsymbol{\varphi}} \beta$ if $r(\alpha) \leq r(\beta)$. We note that if $\{i : \alpha \models \varphi_i\} = \emptyset$, $r(\alpha) = \infty$. Thus, alternatives which do not satisfy any of the formulas in $\boldsymbol{\varphi}$ are less preferred than any other alternative. RPFs generate total preorders.

Revisiting the domain over the power set of $O = \{a, b, c, d\}$, we might define an RPF as $a \wedge b \succ (a \wedge c) \vee (b \wedge d) \succ T$. In this case the alternative $\{a, b, c\} \succ \{a, c, d\}$ given that $\{a, b, c\}$ satisfies the rank 1 formula (and thus is given a rank value of 1) while $\{a, c, d\}$ satisfies the rank 2 formula, thus making $\{a, b, c\}$ more preferred.

We selected these two preference representations because they share a common property, namely the problem of dominance is in P for both representations. Moreover, as we will show, when incomplete alternatives are considered their complexity for N-DOM and P-DOM differs, thus showing the potential change of complexity to dominance problems when incomplete alternatives are introduced.

Results

The introduction of possibly unknown values into alternatives does have the ability to increase the computational complexity of the dominance problem for a preference representation. As one would expect, at best the complexity of P-DOM and N-DOM remains constant with the problem of dominance. In the worst case, we find that the problem has an upper complexity limit which is close, but significant, to the original dominance problem. Results relative to P-DOM are given in Theorems 1 and 2.

Theorem 1. *If DOM($\mathcal{L}$) $\in \Sigma_k^P$ then P-DOM($\mathcal{L}$) $\in \Sigma_{k+1}^P$.*

Proof. By definition the complexity class Σ_{k+1}^P consists of those algorithms that can be checked in polynomial time given a Σ_k^P oracle. This means if we have a DOM($\mathcal{L}$) oracle and an instance of P-DOM($\mathcal{L}$) for a given preference $\succeq$ and two partial alternative α and β then we can guess a completion of α and β and check if $(\alpha, \beta, \succeq) \in$ DOM($\mathcal{L}$) in polynomial time. This means that P-DOM($\mathcal{L}$) is in complexity class Σ_{k+1}^P. □

Theorem 2. *If DOM($\mathcal{L}$) is Σ_k^P-hard then P-DOM($\mathcal{L}$) is Σ_k^P-hard.*

Proof. The class of problems described by P-DOM($\mathcal{L}$) includes all problems described by DOM($\mathcal{L}$) as they are merely instances of P-DOM($\mathcal{L}$) where both alternatives are complete. This means there exists a polynomial time mapping from instances to DOM($\mathcal{L}$) to instances of P-DOM($\mathcal{L}$) where the instance is a member of P-DOM($\mathcal{L}$) iff the instance is also a member of DOM($\mathcal{L}$). Since DOM($\mathcal{L}$) is Σ_k^P-hard, P-DOM($\mathcal{L}$) must also be Σ_k^P-hard. □

The problem of N-DOM is related to the problem of P-DOM and thus we can describe its complexity in terms of P-DOM.

Theorem 3. *If P-DOM($\mathcal{L}$) is in Σ_k^P then N-DOM($\mathcal{L}$) is in Π_k^P if $\mathcal{L}$ produces a total order.*

Proof. The class of problems described by Π_k^P is the class of all problems which are the complement of Σ_k^P. If $\mathcal{L}$ produces a total order then for any two alternatives α and β either $\alpha \succeq \beta$ or $\beta \succeq \alpha$ is true, but not both. As such if we have two incomplete alternatives α' and β' then if there exists no completion such that β dominates α then α must dominate β. In other words, α N-DOMs β is β does not P-DOM α. This means the problem of N-DOM is the complement of P-DOM in the case of total orders, thus if P-DOM is in Σ_k^P then N-DOM is in Π_k^P □

As we have shown it is possible that P-DOM and N-DOM are more computationally complex problems than DOM. These results give an upper limit, but are not tightly bound. This means that to determine the complexity of P-DOM and N-DOM for a particular preference representation one must perform some analysis of the problems for the representation of interest. Here we look at RPFs and LPMs as a case study of this type of analysis. While the complexity of DOM is the same for both representations, the resulting complexity of P-DOM and N-DOM is different. First, we look into the complexity of the P-DOM problem on LPMs, Proposition 1 , and RPFs, Proposition 2.

Proposition 1. *The problem of P-DOM(LPM) is in P.*

Proof. Consider the process for determining dominance between two alternatives α and β wrt an LPM, pi. For each attribute we compare the values between α and β. We may run into the following cases:

- If the values are known and the same, we continue to the next most important attribute.
- If the values are known and different we determine if $\alpha \succ \beta$ and if so then α possibly dominates β and if not then α does not possibly dominate β.
- If the value is only known in β and the value of β is not the optimal value, we can assign α the optimal value for that attribute, thus meaning that α possibly dominates β.
- If the value is only known in β and the value of β is the optimal value, we assign α the optimal value for that attribute and continue to the next most important attribute.
- If the value is only known for α and the value of α is not the pessimal value, we assign β the pessimal value, thus meaning α possibly dominates β.
- If the value is only known for α and the value of α is the pessimal value, we assign β the pessimal value and continue to the next most important attribute.
- If the value is unknown for both α and β we assign α the optimal value and β the pessimal value, thus α possibly dominates β.
- If we exhaust all attributes without determining possible dominance then α does not possibly dominate β because under the optimal choices for α and pessimal for β they are preferentially equivalent.

Since this process only takes time proportional to the number of attributes the determination of possible dominance for LPMs takes polynomial time. □

Proposition 2. *The problem of P-DOM(RPF) is NP-complete.*

Proof. **P-Dom(RPF) $\in$ NP**
Consider an instance of P-Dom(RPF) for a given RPF Ψ and two incomplete alternatives α and β. We can guess completions for both α and β (α' and β', respectively) and compute their ranks according to Ψ. If $\alpha' \succ_\Psi \beta'$ then we know that the instance is in P-Dom(RPF). This process takes polynomial time because computing dominance wrt a RPF takes time polynomial in the size of the RPF. Thus, P-Dom(RPF) $\in$ NP.

P-Dom(RPF) is NP-hard
We show this via a polynomial time mapping reduction from SAT. Given a boolean formula over k variables, Φ, we construct an instance of P-Dom(RPF), $f(\Phi) = (\Psi, \alpha, \beta)$. The domain of Ψ is a binary combinatorial domain with $k+1$ attributes. Each attribute $1 \cdots k$ maps directly onto the same-indexed boolean variable from Φ. Ψ is an RPF constructed such that the ranks are as follows:

1. $\Phi \wedge x_{k+1}$
2. $\neg x_{k+1}$
3. k_{k+1}

We then construct α and β such that all attributes 1 through k for both α and β are unknown. We then set $\alpha[k+1] = 1$ and $\beta[k+1] = 0$.

If $\Phi \in$ SAT then $f(\Phi) \in$ P-Dom(RPF)
If Φ is satisfiable then there exists a setting of variables that makes the formula true. If we take those values and use them to complete α such that if boolean variable x_i is true in the satisfying assignment then $\alpha[i] = 1$ and if not then $\alpha[i] = 0$, then for any satisfying assignment of Φ there exists a completion of α such that α has rank 1. Since $\beta[k+1] = 0$, β can only ever achieve rank 2. Thus, if Φ is satisfiable there exists a completion of α such that $\alpha \succ_\Psi \beta$ meaning $f(\Phi) \in$ P-Dom(RPF).

If $\Phi \notin$ SAT then $f(\Phi) \notin$ P-Dom(RPF)
If Φ is not satisfiable then there exists no setting of variables

that makes the formula true. This means that there is no setting of values in α such that it will be awarded rank 1 by Ψ and thus α will always be given rank 3. Since β will always be given rank 2, there is no completion of α and β such that $\alpha \succ_\Psi \beta$ and thus $f(\Phi) \not in$ P-Dom(RPF).

We have shown that P-Dom(RPF) is both in class NP and NP-hard, thus P-Dom(RPF) is NP-complete. □

As shown above P-DOM is in the same complexity class as DOM for LPMs, but is increased for RPFs. When turning our analysis to N-DOM we see similar results. LPM complexity is unaffected by the introduction of incomplete alternatives, Proposition 3, and RPFs change complexity, Proposition 4. In this case, N-Dom is CoNP complete for RPFs.

Proposition 3. *The problem of N-DOM(LPM) is in P.*

Proof. Consider the process for determining dominance between two alternatives α and β wrt an LPM, pi. For each attribute we compare the values between α and β. We may run into the following cases:

- If the values are known and the same, we continue to the next most important attribute.
- If the values are known and different we determine if $\alpha \succ \beta$ and if so then α necessarily dominates β and if not then α does not necessarily dominate β.
- If the value is only known in β and the value of β is not the pessimal value, then α can be assigned a value such that β dominates α and so α does not necessarily dominate β.
- If the value is only known in β and the value of β is the pessimal value, we assign α the pessimal value for that attribute and continue to the next most important attribute.
- If the value is only known for α and the value of α is not the optimal value, β can be assigned the optimal value, thus meaning α does not necessarily dominate β.
- If the value is only known for α and the value of α is the optimal value, we assign β the optimal value and continue to the next most important attribute.
- If the value is unknown for both α and β we assign α the pessimal value and β the optimal value, thus α does not necessarily dominate β.
- If we exhaust all attributes without determining necessary dominance then α does not necessarily dominate β because under the pessimal choices for α and optimal for β they are preferentially equivalent.

Since this process only takes time proportional to the number of attributes, the determination of necessary dominance for LPMs takes polynomial time. □

Proposition 4. *The problem of N-DOM(RPF) is CoNP-complete.*

Proof. **The problem of not N-DOM(RPF) is in NP**
Given an RPF Φ and two incomplete alternatives α and β, the instance is not a member of N-DOM(RPF) if there exists a completion of α and β (α' and β', respectively) such that $\beta \succeq_\Phi \alpha$. Since the process of determining preference in an RPF is in P, we can guess completions for α and β (α' and β', respectively) and if those completions are such that $\beta' \succeq_\Phi \alpha'$ then α does not necessarily dominate β. Thus the problem of not N-DOM(RPF) is in NP and so it's complement N-DOM(RPF) must be in CoNP. □

The above results show, explicitly, that the complexity of the problems of P-DOM and N-DOM are representation specific, despite placing non-tight limits on the computational complexity based on the dominance problem. Importantly, this limits the required analysis needed to determine the complexity of introducing incomplete alternatives to a particular preference representations.

Discussion and Conclusions

Incomplete alternatives, and its effects on the complexity of related dominance problems, presents a new avenue of analysis for preference representations over combinatorial domains. Of course, dominance is not the only problem of interest when considering preferences. We feel it important to note that since the domain of incomplete alternatives subsumes the set of complete alternatives, the related problems of maximal and minimal preference are unaffected as their solutions would be the same whether or not we allow for incompleteness.

This work also introduces many open questions. At the forefront of our interest is the effects of crossing incomplete alternatives with incomplete preference information, both in terms of unknown and probabilistic information. As a form of analysis resistance to complexity increases due to incomplete alternatives could provide an interesting method of relating various preference representations, similar to the analysis of preference language subsumption (Huelsman and Truszczynski 2022).

Overall, incomplete alternatives represent a new avenue of research for understanding how preferences can be used in real world scenarios. This type of analysis, such as the resilience of a particular preference representation, can help to inform choices about which preference representation is most useful for a given application and how well that representation can handle uncertainty.

References

Cornelio, C.; Goldsmith, J.; Grandi, U.; Mattei, N.; Rossi, F.; and Venable, K. B. 2021. Reasoning with pcp-nets. *Journal of Artificial Intelligence Research* 72:1103–1161.

Cullinan, J.; Hsiao, S. K.; and Polett, D. 2014. A borda count for partially ordered ballots. *Social Choice and Welfare* 42(4):913–926.

Domshlak, C.; Hüllermeier, E.; Kaci, S.; and Prade, H. 2011. Preferences in AI: An overview. *Artificial Intelligence* 175(7–8):1037–1052.

Fagin, R.; Kumar, R.; Mahdian, M.; Sivakumar, D.; and Vee, E. 2006. Comparing partial rankings. *SIAM Journal on Discrete Mathematics* 20(3):628–648.

Fishburn, P. C. 1974. Lexicographic orders, utilities and decision rules: A survey. *Management science* 20(11):1442–1471.

Huelsman, M. A., and Truszczynski, M. 2022. Relating preference languages by their expressive power. In *The International FLAIRS Conference Proceedings*, volume 35.

Kaci, S. 2011. *Working with Preferences: Less is More*. Springer.

Liu, X., and Truszczynski, M. 2015. Learning partial lexicographic preference trees over combinatorial domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.