More than the sum of its parts: compact preference representation over combinatorial domains

Uckelman, J.D.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 1

Introduction

The whole is more than the sum of its parts.

*Metaphysics*, Book VIII, 1045a10

Aristotle

“I would prefer not to.”

*Bartleby, The Scrivener. A Story of Wall-Street*

Herman Melville

The miserable wasteland of multidimensional space was first brought home to me in one gruesome solo lunch hour in one of MIT’s sandwich shops. “Wholewheat, rye, multigrain, sourdough or bagel? Toasted, one side or two? Both halves toasted, one side or two? Butter, polyunsaturated margarine, cream cheese or hummus? Pastami, salmon, lox, honey cured ham or Canadian bacon? Arugula, iceberg, romaine, cress or alfalfa? Swiss, American, cheddar, mozzarella, or blue? Tomato, gherkin, cucumber, onion? Wholegrain, French, English or American mustard? Ketchup, piccalilli, tabasco, soy sauce? Here or to go?”

*Balliol College Annual Record, 2001*

Myles Aston

What are preferences? Why have them over a combinatorial domain? Why do we want to represent them compactly? Why represent them at all? We begin this dissertation by unpacking its subtitle and addressing these questions.

**Compact Preference Representation Over Combinatorial Domains**

Any entity not wholly indifferent to the state of the world has preferences. I prefer ales to pilsners, my cat prefers to be petted in one direction over the other, 131 million people expressed their preference for President of the United States by voting in the 2008 general election [Federal Election Commission, 2009]. These are *ordinal* preferences, ranking one alternative ahead of another. *Cardinal* preferences, which assign values to alternatives, are ubiquitous as well: All monetary transactions involve cardinal preferences as prices. I would pay $300
for a camera with features X and Y, and €2 for a coffee in Paris. Stock markets collect the cardinal preferences of investors; auctions do the same for bidders. It is true (though possibly trite): Preferences are everywhere.

Compact Preference Representation Over Combinatorial Domains

The ubiquity of preferences in our interactions with each other brings about the need for us to express them. When you place a bid in an auction on the Internet auction site eBay, your bid encodes what you are willing to pay for the item being auctioned. My cat bats my hand away when he’d rather I leave him alone. I say to my dinner companions that I would rather eat at the Thai than the Indonesian restaurant. Voters mark ballots for their preferred candidates. In all of these cases, there is a mechanism by which individuals—agents—translate their preferences from whatever form they take inside their heads into a form which is visible to others. This external form is the representation of an agent’s preferences. Representations matter: Arguably, the outcome of the 2000 U.S. Presidential election was due to a faulty preference representation method.

Compact Preference Representation Over Combinatorial Domains

If I have a basket of fruit and offer you a piece, then what I am asking you to do is express your preferences over single pieces of fruit. The domain—that is, the set of alternatives—is the contents of the basket. If my basket contains an apple, a banana, a cherry, a fig, a grapefruit, a lime, a mango, an orange, and a peach, then it will not be overly difficult for you to give your entire preference order over the pieces of fruit. For example, you might say that

\[ G > C > A > P > B > F > N > M > O > L, \]

where > is to be read as “is preferred to”. I might need your full preference order because I am offering fruit to others as well, and I want to ensure that no one is stuck with their last choice. Now consider what happens if I am giving away not just single pieces of fruit, but arbitrary collections of it. Originally, the possible outcomes for you were ten—for each single piece of fruit, you could be given it. Now, the space of potential outcomes has grown exponentially: You could be given any of the 1023 (= 2^{10} – 1) combinations of the pieces of fruit in the basket. We have moved from a simple domain to a combinatorial one.

Compact Preference Representation Over Combinatorial Domains

We continue with the fruit basket example: If I need your complete preference ordering over all 1023 nonempty subsets of fruit in order to make a decision about what fruit to give you, then we are facing a serious problem. You will surely not want to rank each of the 1023 nonempty subsets of the fruit in the basket even
if you are able to do so; moreover, I will not want to wait for you to do it, nor
would I want to deal with such a torrent of information even if you were able to
produce it quickly. If I added another ten pieces of fruit to my basket, the number
of subsets of fruit would already exceed one million, and with 300 pieces of fruit,
there are more subsets \(2^{300} \approx 2 \times 10^{90}\) than atoms in the observable universe
\(\approx 10^{80}\). The problem being described here is known as exponential blowup, and
is a perennial issue when the space of alternatives has a combinatorial structure.
What is needed here is a more compact way of expressing your preferences over
the subsets of fruit, one which does not require you to list your ordering explicitly,
but rather takes advantage of the structure of your preferences and in so doing
permits you to convey them concisely. Even for computerized agents, handling
preferences over combinatorial domains can quickly become unmanageable without
good representations.

Having unpacked the subtitle, it should be clear why compact preference
representation over combinatorial domains is needed. Now on to the title: A little
reflection on common experience reveals that we often have complex preferences,
even over multiples of the same type of thing. For example, while I might be
willing to pay $3 for my first ice cream cone, it is unlikely that I will place as high
a value on a second, third, or fourth cone. At some point, I might even refuse to
accept additional ice cream cones offered to me at no cost. The upshot is that
my—and probably also, your—preferences over ice cream cones are such that I
will value \(n\) cones less than \(n\) times the value I place on one cone. My preferences
are subadditive.

Examples pointing in the opposite direction exist as well: Adjacent plots of
land may be worth more together than individually. Rights to use sections of
railway tracks are more valuable in combination when they link desirable locations.
Matching trucks with truckloads to prevent trucks from traveling empty in one
direction makes the loads more valuable together than singly. Complete sets of
baseball cards are worth more than the individual cards comprising them. Takeoff
and landing rights at airports are useless if not matched. All of these are examples
of superadditive preferences. (For more examples of complex preferences see [Ball,
Donohue, and Hoffman, 2006; Caplice and Sheffi, 2006; Cantillon and Pesendorfer,
2006; Sandholm, 2007].)

What these cases all have in common is that the goods involved interact to
affect their value as a collection. Together, they have value which is more (or less)
than the sum of the values of their constituent parts—and this is where the
difficulty of preference representation over combinatorial domains lies. Modular
preferences, ones where the values of goods are independent from one another, are
simple to represent. There is no need to introduce any conceptual heavy machinery
to handle them, they may be written concisely in an obvious way, and almost
all computational problems involving them are easy. But as soon as we move
away from modular preferences—and, as we have seen, nonmodular preferences
are found abundantly in the real world—we are immediately confronted with representation problems. These are the problems we tackle in this dissertation.

We return once more to the fruit basket example, to give a taste of how the structure of an agent’s preferences can be exploited to dramatically simplify their representation. As mentioned above, modular preferences are easy to represent. If the values of the fruits are independent for some agent, then we can represent his preferences over bundles of fruit by writing the agent’s value for each piece of fruit:

\[ \{(A, 1), (B, 3), (C, 2), (F, 2), (G, 1), (L, 3), (M, 4), (N, 2), (O, 1), (P, 3)\} \]

Then, all we need to do to determine this agent’s value for any bundle of fruit is to sum the values of the individual pieces contained in it. For example, the bundle \{B, F, G\}, containing the banana, the fig, and the grapefruit, has value 6 for an agent with these preferences.

An agent might well have more complex structure to his preferences than this. For example, he might be allergic to cherries, and so any bundle containing cherries is worse than one without cherries. He might intend to cook a dessert which uses both limes and mangoes, but has no use for one without the other. Or he might not care what fruit he gets, so long as he gets at least one piece. Look again at the example of modular preferences over the fruit. We assigned a symbol—a propositional variable—to each piece of fruit, and a value which accrues to the agent for receiving that piece of fruit—for making that propositional variable true. This suggests an extension, by which we use more complex logical formulas instead of just single propositional variables. In this way, we can say \(\neg C\) if we want to avoid cherries, \(L \land M\) if we get extra value from receiving the lime and mango together, and \(A \lor B \lor C \lor F \lor G \lor L \lor M \lor N \lor O \lor P\) to say that we want at least one piece of fruit, but we don’t care which one. So,

\[ \{(-C, 1), (L \land M, 5)\} \]

could be the preferences of an agent who wants to avoid cherries and to get the lime and mango together. These kinds of languages, languages of weighted formulas, are the method of compact preference representation which we pursue in this dissertation.

Chapter Overview

The aim of this dissertation is to explore the possibilities of a particular formalism for compact preference representation over combinatorial domains—sets of weighted formulas, known as goalbases. The overall structure may be seen in Figure 1.1. A chapter at the head of an arrow relies on results from the chapter at the tail of the same arrow.
Chapter 1, Introduction is the chapter you are reading now.

Chapter 2, Languages introduces the basic formalism and notation used throughout this dissertation. In particular, we define goalbases, goalbase languages, and the various restrictions which may be placed on them; we show how goalbases generate utility functions, and thereby represent cardinal preferences; and we give a wide-ranging overview of other preference representation languages, both ordinal and cardinal.

Part I, Theory is the heart of this dissertation, where we explore the properties of the goalbase languages defined in Chapter 2. In particular, we examine in detail the expressivity, succinctness, and complexity of each language.

Chapter 3, Expressivity takes up a basic question about each goalbase language defined in Chapter 2, namely: Which utility functions are expressible in each language? We show that many goalbase languages correspond exactly to classes of utility functions having well-known properties. Along the way, we also prove some results about the variety of available representations in certain languages. In particular, we show that some goalbase languages have exactly one representation for each utility function they are able to represent, a property we call unique representations.

Chapter 4, Succinctness considers how compactly our goalbase languages are able to represent utility functions. Here, we present numerous pairwise
comparisons between goalbase languages, in some cases showing that one language
is exponentially more succinct than another. Due to our systematic approach, this
chapter contains hundreds of results, conveniently summarized in several tables.

Chapter 5, Complexity classifies goalbase languages according to the computa-
tional complexity of deciding various questions concerning goalbases in those
languages. For many (though not all) goalbase languages, the decision problem
max-util, which asks whether an alternative exists which produces at least a
given level of utility, is \textbf{NP}-complete. Similarly, the problem min-util, which
asks whether all alternatives yield at least some minimum amount of utility, is
\textbf{coNP}-complete for many of the more expressive languages. Thirdly, we consider
the problem max-cuf, which deals with maximizing collective utility, rather
than individual utility as max-util does, and again find that for many—though,
significantly, not all—goalbase languages, max-cuf is \textbf{NP}-complete. Finally, we
consider an alternative version of max-util, which asks about true atoms in
optimal states instead of the existence of states yielding at least a given amount
of utility.

The chapters in Part I are based on and extend work presented at the AAAI-
2007 Workshop on Preference Handling for Artificial Intelligence (AiPref-2007)
[Uckelman and Endriss, 2007], the 11th International Conferences on Principles
of Knowledge Representation and Reasoning (KR-2008) [Uckelman and Endriss,
2008b], and in the journal article “Representing Utility Functions via Weighted
Goals” [Uckelman, Chevaleyre, Endriss, and Lang, 2009]. In turn, the latter
includes some results due to Chevaleyre, Endriss, and Lang [2006]. Section 5.7
extends work presented at the AAAI-2008 4th Multidisciplinary Workshop on
Advances in Preference Handling (MPREF-2008) [Uckelman and Witzel, 2008].

Part II, Applications highlights two areas in which goalbase languages may
be used to good effect.

Chapter 6, Auctions is the first of our two chapters showing applications of
goalbase languages. Auctions are a common way of selling goods. Unfortunately,
sequential auctions—auctions where individual goods are sold consecutively—are
inefficient when the values of goods being sold are interdependent. Combinatorial
auctions are a method of auctioning all goods simultaneously, so that synergies
among goods may be taken into account. In this chapter, we discuss existing
bidding languages for combinatorial auctions, suggest the use of goalbase languages
for bids, present two algorithms for solving the Winner Determination Problem
for combinatorial auctions when using goalbases as bids, and give experimental
results for these algorithms.
All save Section 6.3 of this chapter is based on and extends work presented at the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2008) [Uckelman and Endriss, 2008a].

Chapter 7, Voting points toward elections as another area in which goalbase languages may be useful. We consider two main problems: First, many voting methods are insufficiently expressive to capture the preferences of voters. Second, voting methods intended to choose only a single winner acquire undesirable properties when modified to produce multiple winners, as in elections for committees. In this chapter, we argue that using goalbases as ballots has potential both as a way of extending the expressivity of single-winner voting methods, and also as a way of handling the combinatorial nature of voters’ preferences in multi-winner voting.

Chapter 8, Conclusion is at the end, summarizing what we have shown and suggesting some avenues for further work.