AI Magazine Volume 28 Number 4 (2007) (© AAAI)

Representing and Reasoning with Preferences

Toby Walsh

■ I consider how to represent and reason with users' preferences. While areas of economics like social choice and game theory have traditionally considered such topics, I will argue that computer science and artificial intelligence bring some fresh perspectives to the study of representing and reasoning with preferences. For instance, I consider how we can elicit preferences efficiently and effectively.

Preferences are a central aspect of decision making for single or multiple agents. With one agent, the agent's desired goal may not be feasible. The agent wants a cheap, low-mileage Ferrari, but no such car exists. We may therefore look for the most preferred outcome among those that are feasible. With multiple agents, their goals may be conflicting. One agent may want a Prius, but another wants a Hummer. We may therefore look for the outcome that is most preferred by the agents. Preferences are thus useful in many areas of artificial intelligence including planning, scheduling, multiagent systems, combinatorial auctions, and game playing.

Artificial intelligence is not the only discipline in which preferences are of interest. For instance, economists have also studied preferences in several contexts including social choice, decision theory, and game theory. In this article, I will focus on the connections between the study of preferences in artificial intelligence and in social choice. Social choice is the theory of how individual preferences are aggregated to form a collective decision. For example, one person prefers Gore to Nader to Bush, another prefers Bush to Gore to Nader, and a third prefers Nader to Bush to Gore. Who should be elected? There are many useful ideas about preferences that have been imported from social choice into artificial intelligence. For example, as I will discuss later in this article, voting procedures have been proposed as a general mechanism to combine agents' preferences. As a second example, ideas from game theory like Nash equilibrium have proven very influential in multiagent decision making.

In the reverse direction, artificial intelligence brings a fresh perspective to some of the questions addressed by social choice. These new perspectives are both computational and representational. From a computational perspective, we can look at how computationally we reason with preferences. As we shall see later in this article, computational intractability may actually be advantageous in this setting. For example, we can show that for a number of different voting rules manipulating the result of an election is possible in theory, but computationally difficult to perform in practice. From a representational perspective, we can look at how we represent preferences, especially when the number of outcomes is combinatorially large. We shall see situations where we have a few agents but very large domains over which they are choosing.

Another new perspective, both computational and representational, is how we represent and reason about uncertainty surrounding preferences. As we shall see, uncertainty can arise in many contexts. For example, when eliciting an agent's preferences, we will have uncertainty about some of their preferences. As a second example, when trying to manipulate an election, we may have uncertainty about the other agents' votes. As a third example,

 ISSN 0738-4602

there may be uncertainty in how the chair will perform the election. For instance, in what order will the chair compare candidates? Such uncertainty brings fresh computational challenges. For example, how do we compute whether we have already elicited enough preferences to declare the winner?

Representing Preferences

As with other types of knowledge, many different formalisms have been proposed and studied to represent preferences. One broad distinction is between cardinal and relational preference representations. In a cardinal representation, a numerical evaluation is given to each outcome. Such an evaluation is often called a utility. In a relational representation, on the other hand, a ranking of outcomes is given by means of a binary preference relation. For example, we might simply have that the agent prefers a "hybrid car" to a "diesel car" without assigning any weights to this. In the rest of the article, I shall restrict much of my attention to this latter type of relational representation. It is perhaps easier to elicit relational preferences: the agent simply needs to be able to rank outcomes. In addition, it is perhaps easier to express conditional choices using a relational formalism. For example, if the car is new, the agent might prefer a "hybrid car" to a "diesel car," but if the car is secondhand, the agent is concerned about battery replacement and so prefers a "diesel car" to a "hybrid car." Such a conditional preference is difficult to express using utilities but, as we shall see, is straightforward with certain relational formalisms. Nevertheless, utilities have an important role to play in representing agents' preferences.

A binary preference relation is generally assumed to be transitive. That is, if the agent prefers a "hybrid car" to a "diesel car," and a "diesel car" to a "petrol car" then the agent also prefers a "hybrid car" to a "petrol car." There are three other important properties to consider: indifference, incompleteness, and incomparability. It is important to make a distinction between these three "I"s. Indifference represents that the agent likes two outcomes equally. For instance, the agent might have an equal (dis)like for sports utilities and minivans. Incompleteness, on the other hand, represents a gap in our knowledge about the agent's preferences. For instance, when eliciting preferences, we may not have queried the agent yet about its preference between an "electric car" and a "hybrid car." We may wish to represent that the preference relation is currently incomplete and that at some later point the precise relationship may become known. Finally, incomparability represents that two outcomes cannot in some fundamental sense be compared with each other. For example, an agent might prefer a "hybrid car" to a "diesel car" and a "cheap car" to an "expensive car." But the agent might not want to compare an "expensive hybrid" with a "cheap diesel." A cheap diesel has one feature that is better (the price) but one feature that is worse (the engine). The agent might want both choices to be returned, as both are Pareto optimal (that is, there is no car that is more preferred). Such incomparability is likely to arise when outcomes combine together multiple features. However, it also arises when we are comparing outcomes that are essentially very different (for example, a car and a bicycle).

Another important aspect of representing preferences is dealing with the combinatorial nature of many domains. Returning to the car domain, we have the engine type, the number of seats, the manufacturer, the age, as well as the price and other features like the fuel consumption, the color, the trim, and so on. A number of formalisms have been proposed to represent preferences over such large combinatorial domains. For example, CP-nets decompose a complex preference relation into conditionally independent parts (Boutilier et al. 1999). CP-nets exploit the ceteris paribus ("all else being equal") assumption under which the preference relation depends only on features that change. This formalism lets us represent preferences over a complex feature space using a small number of possibly conditional preference statements (see figure 1). CP-nets exploit conditional independence within the preference relation in much the same way as a Bayes network tries to compactly represent a complex probability distribution function.

A number of extensions of CP-nets have been proposed including TCP-nets to represent trade-off (for example, "price" is more important to me than "engine type") (Brafman and Domshlak 2002) and mCP-nets to represent the preferences of multiple agents (each agent has its own CP-net and these are combined using voting rules) (Rossi, Venable, and Walsh 2004). However, there are several outstanding issues concerning CP-nets including their decisiveness and their complexity. CP-nets induce in general only a partial order over outcomes (such as the "expensive hybrid" versus "cheap diesel" example). A CP-net may not therefore order enough outcomes to be useful in practice. In addition, reasoning with CP-nets is in general computationally hard. For example, determining whether one outcome is more preferred than another is PSPACE-hard (Goldsmith et al. 2005). To reduce this complexity, various approximations have been proposed (Domshlak et al. 2003, 2006). In addition, restricted forms of CP-nets have been identified (for example, those where the dependency between features is acyclic) where reasoning is more tractable (Boutilier et al. 1999). However, more work needs to be done if CP-nets are to find application in practical systems.

Another way to represent an agent's preferences is by means of the agent's ideal and nonideal outcomes. For instance, an agent might like two cars on display (say, a Volvo and a Jaguar) but not the third car (say, a Lada). The agent might therefore specify "I would like something like the Volvo or the Jaguar but not the Lada." Hebrard, O'Sullivan, and Walsh (2007) proposed a method to reason about such logical combinations of ideal and nonideal outcomes (see figure 2 for some more details). This approach has a flavor of qualitative methods, in allowing logical combinations of ideals and nonideals, and quantitative methods, in measuring distance from such ideals and nonideals. The propagators developed to reason about such distance constraints are, it turns out, closely related to those that can return a diverse set of solutions ("show me five different cars that satisfy my constraints") and those that return a similar set of solutions ("show me some similar cars to this one that also satisfy my constraints") (Hebrard et al. 2005). An attractive feature of representing preferences through ideal and nonideal outcomes is that preference elicitation may be quick and easy. Agents need answer only a few questions about their preferences. On the downside, it is difficult to express the sort of complex conditional preferences that are easy to represent with formalisms like CP-nets. An interesting research direction would be to learn (conditional) preference statements like those used in CP-nets given some ideal and nonideal outcomes.

Preference Aggregation

In multiagent systems, we may need to combine the preferences of several agents. For instance, each member of a family might have preferences about what car to buy. A common mechanism for aggregating together preferences is to apply a voting rule. Each agent expresses a preference ordering over the set of outcomes, and an election is held to compute the winner. When there are only two possible outcomes, it is easy to run a "fair" election. We

Suppose an agent declares unconditionally that a hybrid is better than a diesel. We write this:

hybrid > *diesel*

With a hybrid, the agent declares:

saloon > *station wagon*

But with a diesel, the agent declares:

station wagon > *saloon*

Then a *hybrid saloon* is more preferred to a *hybrid station wagon* since *ceteris paribus* we keep the engine type constant but move from the more preferred *saloon* to the less preferred *station wagon* according to the first conditional preference statement: with a *hybrid* then *saloon* > *station wagon.*

A *hybrid station wagon* itself is more preferred to a *diesel station wagon* since we move from the more preferred *hybrid* to the less preferred *diesel* according to the first (unconditional) preference statement: *hybrid* > *diesel.*

Finally a *diesel station wagon* is more preferred to a *diesel saloon* since we keep the engine type constant but move from the more preferred station wagon to the less preferred saloon according to the second conditional preference statement: with a *diesel* then *station wagon* > *saloon.*

Thus, we have:
hybrid saloon > *hybrid station wagon* >
diesel station wagon > *diesel saloon.*

Figure 1. An Example CP-Net.

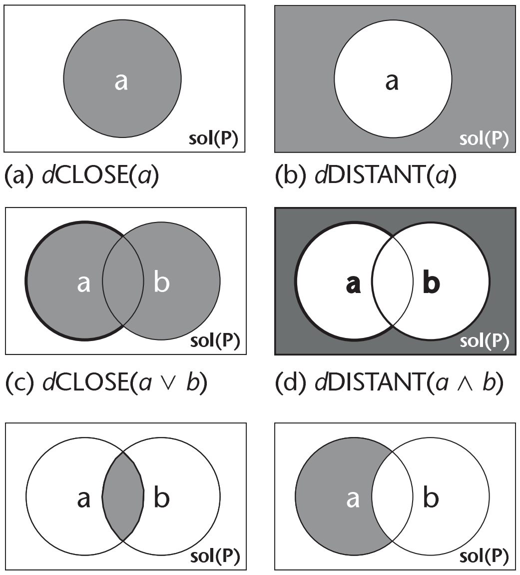
We suppose the user expresses her preferences in terms of ideal or nonideal (partial) solutions. Partiality is important so we can ignore irrelevant attributes. For example, we might not care whether our ideal car has run-flat tires or not.

One of the fundamental decision problems underlying this approach is that a solution is at a given distance d to (resp. from) an ideal (resp. nonideal) solution. These are called dCLOSE and dDISTANT respectively.

We can specify more complex preferences by using negation, conjunction, and disjunction.

$$\begin{aligned} d\text{DISTANT}(a) &\leftrightarrow \neg d\text{CLOSE}(a) \\ d\text{DISTANT}(a \vee b) &\leftrightarrow d\text{DISTANT}(a) \vee d\text{DISTANT}(b) \\ &\leftrightarrow \neg d\text{CLOSE}(a \wedge b) \\ d\text{DISTANT}(a \wedge b) &\leftrightarrow d\text{DISTANT}(a) \wedge d\text{DISTANT}(b) \\ &\leftrightarrow \neg d\text{CLOSE}(a \vee b) \end{aligned}$$

These can be represented graphically: a and b are solutions, $sol(P)$ is the set of solutions within the distance d, and the shaded region represents the solutions that satisfy the constraints.



(a) dCLOSE(a) (b) dDISTANT(a)

(c) dCLOSE($a \vee b$) (d) dDISTANT($a \wedge b$)

(e) dCLOSE($a \wedge b$) (f) dCLOSE(a) $\wedge$ dDISTANT(b)

Figure 2. Representing Preferences through Ideal and Nonideal Outcomes (Hebrard, O'Sullivan, and Walsh 2007).

apply the majority rule and select the outcome with the most votes. However, elections are more probematic when there are more than two possible outcomes.

Going back to at least the Marquis de Condorcet in 1785, and continuing with Arrow, Sen, Gibbard, Satterthwaite, and others from the 1950s onwards, social choice theory has identified fundamental issues that arise in running elections with more than two outcomes (see figure 3 for an illustrative example). For instance, Arrow's famous impossibility theorem shows that there is no "fair" method to run an election if we have more than two outcomes. Fairness is defined in an axiomatic way by means of some simple but desirable properties like the absence of a dictator (that is, an agent whose vote is the result) (Arrow 1970). A closely related result, the Gibbard-Satterthwaite theorem shows that all "reasonable" voting rules are manipulable (Gibbard 1973; Satterthwaite 1975). The assumptions of this theorem are again not very strong. For example, we have three or more outcomes, and there is some way for every candidate to win (for example, the election cannot be rigged so Gore can never win). Manipulation here means that an agent may get a result it prefers by voting tactically (that is, declaring preferences different to those the agent has). Consider, for instance, the plurality rule under which the outcome with the most votes wins. Suppose you prefer a hybrid car over a diesel car, and a diesel car over a petrol car. If you know that no one else likes hybrid cars, you might vote strategically for a diesel car, as your first choice has no hope.

Strategic voting is generally considered undesirable. There are many reasons for this, including the result is not transparent to the electorate, agents need to be sophisticated and informed to get a particular result, and fraud may be difficult to detect if the result is hard to predict. To discuss manipulability results in more detail, we need to introduce several different voting rules. A vote is one agent's ranking of the outcomes. For simplicity, we will assume this is a total order, but as we observed earlier, it may be desirable in some situations to consider partial orders. A voting rule is then simply a function mapping a set of votes onto one outcome,[1] the winner. We shall normally assume that any rule takes polynomial time to apply. However, there are some voting rules where it is NP-hard to compute the winner (Bartholdi, Tovey, and Trick 1989b). Finally, we will also consider weighted votes. Weights will be integers so a weighted vote can be seen simply as a number of agents voting identically.

Weighted voting systems are used in a number of real-world settings like shareholder meetings and elected assemblies. Weights are useful in multiagent systems that have different types of agents. Weights are also interesting from a computational perspective. For example, adding weights to the votes may introduce computational complexity. For instance, manipulation can become NP-hard when weights are added (Conitzer and Sandholm 2002). As a second example, as I discuss later in the article, the weighted case informs us about the unweighted case when there is uncertainty about the votes. I now define several voting rules that will be discussed later in this article.

Scoring Rules

Scoring rules are defined by a vector of weights, $(w_1, \ldots, w_m)$. The *i*th outcome in a total order scores w_i, and the winner is the outcome with the highest total score. The plurality rule has the weight vector (1, 0, …, 0). In other words, each highest ranked outcome scores one point. When there are just two outcomes, this degenerates to the majority rule. The veto rule has the weight vector (1, 1, …, 1, 0). The winner is the outcome with the fewest "vetoes" (zero scores). Finally, the Borda rule has the weight vector $(m - 1, m - 2, \ldots, 0)$. This attempts to give the agent's second and lower choices some weight.

Cup (or Knockout) Rule

The winner is the result of a series of pairwise majority elections between outcomes. This exploits the fact that when there are only two outcomes, the majority rule is a fair means to pick a winner. We therefore divide the problem into a series of pairwise contests. The agenda is the schedule of pairwise contests. If each outcome must win the same number of majority contests to win overall, then we say that the tournament is balanced.

Single Transferable Vote (STV) Rule

The single transferable vote rule requires a number of rounds. In each round, the outcome with the least number of agents ranking them first is eliminated until one of the remaining outcomes has a majority.

We will also consider one other common preference aggregation rule in which voters do not provide a total ordering over outcomes, but simply a set of preferred outcomes.

Approval Rule

The agents approve of as many outcomes as they wish. The outcome with the most approvals wins.

Consider an election in which Alice votes:

Prius > Civic Hybrid > Tesla

Bob votes:

Tesla > Prius > Civic Hybrid

And Carol votes:

Civic Hybrid > Tesla > Prius

Then we arrive in the "paradoxical" situation where two thirds prefer a Prius to a Civic Hybrid, two thirds prefer a Tesla to a Prius, but two thirds prefer a Civic Hybrid to a Tesla. The collective preferences are cyclic.

There is no fair and deterministic resolution to this example since the votes are symmetric. With majority voting, each car would receive one vote. If we break the three-way tie, this will inevitably be "unfair," favoring one agent's preference ranking over another's.

Figure 3. Condorcet's Paradox.

Manipulation

Gibbard-Satterthwaite's theorem proves that all "reasonable" voting rules are manipulable once we have more than two outcomes (Gibbard 1973, Satterthwaite 1975). That is, voters may need to vote strategically to get their desired result. Researchers have, however, started to consider computational issues surrounding strategic voting and such manipulation of elections. One way around Gibbard-Satterthwaite's theorem may be to exploit computationally complexity. In particular, we might look for voting rules that are manipulable but where the manipulation is computationally difficult to find (Bartholdi, Tovey, and Trick 1989a). As with cryptography, computational complexity is now wanted and is not a curse. For example, it has been proven that it is NP-hard to compute how to manipulate the STV rule to get a particular result if the number of outcomes and agents is unbounded (Bartholdi and Orlin 1991).

One criticism made of such results by researchers from social choice theory is that, while elections may have a lot of agents voting, elections often only choose between a small number of outcomes. However, as argued before, in artificial intelligence, we can have combinatorially large domains. In addition, it was subsequently shown that STV is NP-hard to manipulate even if the number of outcomes is bounded, provided the votes are weighted (Conitzer and Sandholm 2002). Manipulation now is no longer by one strategic agent but by a coalition of agents. This may itself be a more useful definition of manipulation. A single agent can rarely change the outcome of many elections. It may therefore be more meaningful to consider how a coalition might try to vote strategically.

Many other types of manipulations have been considered. One major distinction is between destructive and constructive manipulation. In constructive manipulation, we are trying to ensure a particular outcome wins. In destructive manipulation, we are trying to ensure a particular outcome does not win. Destructive manipulation is at worse a polynomial cost more difficult than constructive manipulation provided we have at most a polynomial number of outcomes. We can destructively manipulate the election if and only if we can constructively manipulate some other outcome. In fact, destructive manipulation can sometimes be computationally easier. For instance, the veto rule is NP-hard to manipulate constructively but polynomial to manipulate destructively (Conitzer, Sandholm, and Lang 2007). This result may chime with personal experiences on hiring committees: it is often easier to ensure someone is not hired than to ensure someone else is!

Another form of manipulation is of individual preferences. We might, for example, be able to manipulate certain agents to put Boris in front of Nick, but Ken's position on the ballot will remain last.[2] Surprisingly, manipulation of individual preferences is more computationally difficult than manipulation of the whole ballot. For instance, for the cup rule, manipulating by a coalition of agents is polynomial (Conitzer, Sandholm, and Lang 2007), but manipulation of individual preferences of those agents is NP-hard (Walsh 2007a).

Another form of manipulation is of the voting rule itself. Consider again the cup rule. This rule requires an agenda, the tree of pairwise majority comparisons. The chair may try to manipulate the result by choosing an agenda that gives a desired result. If the tournament is unbalanced, then it is polynomial for the chair to manipulate the election. However, we conjecture that it is NP-hard to manipulate the cup rule if the tournament is required to be balanced (Lang et al. 2007). Many other types of manipulations have also been considered. For example, the chair may try to manipulate the election by adding or deleting outcomes, adding or deleting agents, partitioning the candidates into two and running an election in the two halves, and so on.

Uncertainty

One important consideration is the impact of uncertainty on voting. One source of uncertainty is in the votes. For example, during preference elicitation, not all agents may have expressed their preferences. Even if all agents have expressed their preferences, a new outcome might be introduced. To deal with such situations, Konczak and Lang (2005) have considered how to reason about voting when preferences are incompletely specified. For instance, how do we compute whether a certain outcome can still win? Can we compute when to stop eliciting preferences? Konczak and Lang introduced the concept of possible winners, those outcomes that win in some transitive completion of the votes, and necessary winner, that outcome that wins in all transitive completions of the votes. Preference elicitation can stop when the set of possible winners equals the necessary winner. Unfortunately, computing the possible and necessary winners is NP-hard in general (Pini et al. 2007). In fact, it is even NP-hard to compute these sets approximately (that is, to within a constant factor in size) (Pini et al. 2007). However, there are a wide range of voting rules, where possible and necessary winners are polynomial to compute. For example, possible and necessary winners are polynomial to compute for any scoring rule (Konczak and Lang 2005).

Another source of uncertainty is in the voting rule itself. For example, uncertainty may be deliberately introduced into the voting rule to make manipulation computationally difficult (Conitzer and Sandholm 2002). For instance, if we randomize the agenda used in the cup rule, the cup rule goes from being polynomial to manipulate to NP-hard. There are other forms of uncertainty that I shall not consider here. For example, preferences may be certain, but the state of the world uncertain (Gajdos et al. 2006). As a second example, we may have a probabilistic model of the user's preference that is used to direct preference elicitation (Boutilier 2002).

Weighted Votes

An important connection is between weighted votes and uncertainty. Weights permit manipulation to be computationally hard even when the number of outcomes is bounded. If votes are unweighted and the number of outcomes is bounded, then there is only a polynomial number of different votes. Therefore to manipulate the election, we can try out all possible manipulations in polynomial time. We can make manipulation computationally intractable by permitting the votes to have weights (Conitzer and Sandholm 2002; Conitzer, Lang, and Sandholm 2003). As I mentioned, certain elections met in practice have weights. However, weights are also interesting as they inform the case where we have uncertainty about how the other agents will vote. In particular, Conitzer and Sandholm proved that if manipulation with weighted votes is NP-hard then manipulation with unweighted votes but a probability distribution over the other agents' votes is also NP-hard (Conitzer and Sandholm 2002).

Preference Elicitation

One interesting application for computing the set of possible and necessary winners is for preference elicitation. The basic idea is simple. Preference elicitation can focus on resolving the relationship between possible winners. For instance, which of these two possible winners is more preferred? Pini et al. (2007) gives a simple algorithm for preference elicitation that focuses elicitation queries on just those outcomes that are possible winners. In fact, under some simple assumptions on the voting rule, the winner can be determined with a number of preference queries that is polynomial in the worst case in the number of agents and outcomes. In practice, we hope it may even be less.

Preference elicitation is closely related to manipulation. Suppose we are eliciting preferences from a set of agents. Since preference elicitation can be time-consuming and costly, we might want to stop eliciting preferences as soon as we can declare the winner. This might be before all votes had been collected. How do we compute when we can stop? If we can still manipulate the election, then the winner is not fixed. However, if we can no longer manipulate the election, the winner is fixed and elicitation can be terminated. It thus follows that manipulation and deciding whether preference elicitation can be terminated are closely related problems. Indeed, if manipulation is NP-hard then deciding whether we can terminate elicitation is also (Konczak and Lang 2005).

Complexity considerations can be used to motivate the choice of a preference elicitation strategy. Suppose that we are combining preferences using the cup rule. Consider two different preference elicitation strategies. In the first, we ask each agent in turn for the agent's vote (that is, we elicit whole votes). In the second, we pick a pair of outcomes and ask all agents to order them (that is, we elicit individual preferences). Then it is polynomial to decide whether we can terminate elicitation using the first strategy, but NP-hard using the second (Walsh 2007a). Thus, there is reason to prefer the first strategy in which we ask each agent in turn for the agent's vote. In fact, many of the manipulation results cited in this paper can be transformed into a similar result about preference elicitation.

Single-Peaked Preferences

One of the concerns with results like those mentioned so far is that NP-hardness is only a worst-case analysis. Are the votes met in practice easier to reason about? For instance, votes met in practice are often single peaked. That is, outcomes can be placed in a left to right order, and an agent's preference for an outcome decreases with distance from the agent's peak. For instance, an agent might have a preferred cost, and the preference decreases with distance from this cost. Single-peaked preferences are interesting from several other perspectives. First, single-peaked preferences are easy to elicit. For instance, we might simply ask you for your optimal house price. Conitzer has given a simple strategy for eliciting a complete preference ordering with a linear number of pairwise ranking questions under the assumption that preferences are single peaked (Conitzer 2007). Second, single-peaked preferences are easy to aggregate. In particular, there is a fair way to aggregate single-peaked preferences. We simply select the median outcome; this is a Condorcet winner, beating all others in pairwise comparisons (Black 1948). Third, with single-peaked preferences, preference aggregation is strategy proof. There is no incentive to misreport preferences.

Suppose we assume that agents' preferences will be single peaked. Does this make it easier to decide whether elicitation can be terminated? We might, for example, stop eliciting preferences if an outcome is already guaranteed to be the Condorcet winner. We suppose we know in advance the ordering of the outcomes that make agents' preferences single peaked. For instance, if the feature is price in dollars, we might expect preferences to be single peaked over the standard ordering of integers. However, an interesting extension is when this ordering is not known. If votes are single peaked and the voting rule elects the Condorcet winner, deciding whether we can terminate elicitation, along with related questions like manipulation, is polynomial (Walsh 2007b).

However, there are several reasons why we might not want to select the Condorcet winner when aggregating single-peaked preferences. For example, the Condorcet winner does not consider the agents' intensity of preferences. We might want to take into account the agents' lower ranked outcomes using a method like Borda. There are also certain situations where we cannot identify the Condorcet winner. For instance, we may not know each agent's most preferred outcome. Many web search mechanisms permit users to specify just an approved range of prices (for example, upper and lower bound on price). In such a situation, it might be more appropriate to use approval voting (which may not select the Condorcet winner) to aggregate preferences. Finally, we might not be able to select the Condorcet winner even if we can identify it. For example, we might have hard constraints as well as preferences. As a result, the Condorcet winner might be infeasible. We might therefore consider a voting system which that returns not just a single winner but a total ranking over the outcomes[3] so that we can return those feasible outcomes which that are not less preferred than other feasible outcomes (so so-called "undominated feasible outcomes"). However, using other voting rules requires care. For instance, manipulation is NP-hard when preferences are single peaked for a number of common voting rules including STV (Walsh 2007b).

Some Negative (Positive?) Results

Various researchers have started to address concerns that NP-hardness is only a worst-case analysis, and votes met in practice might be easier to reason about. For instance, even though it may be NP-hard to compute how to manipulate the STV rule in theory, it might be easy for the sort of elections met in practice. Results so far have been largely negative. That is, manipulation stops being computationally hard. However, in this case, preference elicitation becomes polynomial so these results might also be seen in a positive light! For instance, Conitzer and Sandholm have proven that, with any weakly monotone voting rule,[4] a manipulation can be found in polynomial time if the election is such that the manipulator can make either of exactly two outcomes win (Conitzer and Sandholm 2006). As a second example, Procaccia and Rosenschein have shown that for any scoring rule, you are likely to find a destructive manipulation in polynomial time for a wide class of probability distribution of preferences (Procaccia and Rosenschein 2007). This average case includes preferences that are drawn uniformly at random.

It is perhaps too early to draw a definite conclusion yet about this line of research. A general impression is that while voting rules with a single round may be easy on average, rules with multiple rounds (like STV or the cup rule) introduce a difficult balancing problem. If we are to find a manipulation, we may need to make an outcome good enough to get through to the final round but bad enough to lose the final contest. This may make manipulation computationally difficult in practice and not just in the worst case.

Hybrid Voting Rules

To demonstrate that rules with multiple rounds may be more difficult to manipulate, we consider some recent results on constructing hybrid voting rules. In such rules, we perform some number of rounds of one voting rule (for example, one round of the cup rule or one round of STV) and then finish the election with some other rule (for example, we then complete the election by applying the plurality rule to the remaining outcomes). Consider, for instance, the plurality rule. Like all scoring rules, this is polynomial to manipulate (Conitzer and Sandholm 2002). However, the hybrid rule where we apply one round of the cup rule and then the plurality rule to the remaining outcomes is NP-hard to manipulate (Conitzer and Sandholm 2003). As a second example, the hybrid rule, which has some fixed number of rounds of STV then completes the election with the plurality rule (or the Borda or cup rules), is NP-hard to manipulate (Elkind and Lipmaa 2005). Hybridization does not, however, always ensure computational complexity. For example, the hybrid rule, which has some fixed number of rounds of plurality (in each we eliminate the lowest scoring outcome) and then completes the election with the Borda (or cup) rule, is polynomial to manipulate (Elkind and Lipmaa 2005). Nevertheless, introducing some qualifying rounds to a voting rule seems a good route to some additional complexity, making it computationally difficult to predict the winner or to manipulate the result. It is interesting to wonder if FIFA and other sporting bodies take this into account when deciding the format of major sporting events like the World Cup.

Preferences and Constraints

So far, we have largely ignored the fact that there may be hard constraints preventing us from having the most preferred outcome. For example, we might prefer a cheap car to an expensive car, and a Tesla to a Prius. However, there are no cheap Teslas for us to purchase, at least for the near future. Combining (relational) preferences and constraints in this way throws up a number of interesting computational challenges. One possibility is simply to use the preferences to guide search for a feasible outcome by enumerating outcomes in the constraint solver in preference order (Boutilier et al. 1999). Another possibility is to turn qualitative preferences into soft constraints (Domshlak et al. 2003, 2006). We can then apply any standard (soft) constraint solver. Prestwich et al. (2005), we give a third possibility, a general algorithm for finding the feasible Pareto optimal[5] outcomes, that combines both constraint solving and preference reasoning. The algorithm works with any preference formalism that generates a preorder over the outcomes (for example, it works with CP-nets). Briefly, we first find all outcomes that are feasible and optimal in the preference order. If all the optimals in the preference order are feasible then there are no other feasible Pareto optimals, and we can stop. Otherwise, we must compare these with the other feasible outcomes in case some of these are also feasible Pareto optimals. These three examples illustrate some of the different methods proposed to reason with both constraints and preferences. However, much remains to be investigated.

	Agent1	Agent2	Agent3
1	Y Y Y	Y N N	N Y N
2	Y N N	Y Y Y	N Y Y
3	N Y N	N N N	N N N
4	N N N	N N Y	N Y N
5	Y N Y	Y N Y	Y N Y
6	N Y Y	N Y Y	Y Y Y
7	N N Y	N N Y	Y N N
8	Y Y N	Y Y N	Y Y N

Figure 4. Agents' Preferences over the Three Issues.

Multiple Elections

Another interesting topic is that agents may be expressing preferences over several related issues. This can lead to paradoxical results where multiple elections result in the least favorite combination of issues being decided (Brams, Kilgour, and Zwicker 1998; Lacy and Niou 2000; Xia, Lang, and Ying 2007). For example, suppose agents are deciding their position with respect to three topical issues. For simplicity, I will consider three agents and three binary issues. The issues are: is global warming happening, does this have catastrophic consequences, and should we act now. A position on the three issues can be expressed as a triple. For instance, "N Y Y" represents that global warming is not happening, global warming would have catastrophic consequences, and that we do need to act now. The agents' preferences over the three issues are shown in figure 4.

All agents believe that if global warming is happening and this is causing catastrophic consequences, we must act now. Therefore they all place "Y Y N" last in their preference rankings. As there are an exponential number of outcomes in general, it may be unrealistic for agents to provide a complete ranking when voting. One possibility is for the agents just to declare their most preferred outcomes. As issues are binary, we apply the majority rule to each issue. In this case, this gives "Y Y N." Unfortunately, this is everyone's least favorite option. Lacy and Niou show that such "paradoxical" results are a consequence of the agents' preferences not being separable (Lacy and Niou 2000). We can define the possible and necessary winners for such an election. Both are polynomial to compute. Even though there can be an exponential number of possible winners, the set of possible winners can always be represented in linear space as the issues are decided independently. For example, if Agent1 and Agent2 have voted, but Agent3 has not, the possible winners under the majority rule are "Y* *."

One way around such paradoxical results is to vote sequentially, issue by issue. Suppose the agents vote sincerely (that is, declaring their most preferred option at each point consistent with the current decisions). Here, the agents would decide Y for global warming, then N for catastrophic consequences (as both Agent2 and Agent3 prefer this given Y for global warming), and then N for acting now. Thus, the winner is "Y N N." Lacy and Niou prove that if agents vote sincerely, such sequential voting will not return an outcome dominated by all others (Lacy and Niou 2000). However, such sequential voting does not necessarily return the Condorcet winner, the outcome that will beat all others in pairwise elections. Here, for example, it does not return the Condorcet winner, "Y Y Y." We can define possible and necessary winners for such sequential voting. However, it is not at all obvious whether the possible or necessary winners of such a sequential vote can be computed in polynomial time, nor even whether the set of possible winners can be represented in polynomial space.

Another way around this problem is sophisticated voting. Agents need to know the preferences of the other voters and anticipate the outcome of each vote to eliminate dominated strategies where a majority prefer some other outcome. Lacy and Niou (2000) prove that such sophisticated voting will always produce the overall Condorcet winner if it exists, irrespective of whether voters have separable or inseparable preferences. However, this may not be a computationally feasible solution since it requires reasoning about exponentially large rankings. Another way around this problem is a domain restriction. For instance, Lacy and Niou prove that if agents' preferences are separable, then sequential majority voting is not manipulable, and it is in the best interest of agents to vote sincerely (Lacy and Niou 2000).

Conclusion

This survey has argued that there are many interesting issues concerning the representation of and the reasoning about preferences. Whilst researchers in areas like social choice have studied preferences, artificial intelligence brings some fresh perspectives. These perspectives include both computational questions like the complexity of eliciting preferences and representational questions like dealing with uncertainty. This remains a very active research area. At AAAI-07, there was an invited talk, several technical talks, a tutorial, and a workshop on preferences. It is sure therefore that there will be continuing progress in understanding how we represent and reason with preferences.

Acknowledgements

This article is loosely based on an invited talk given at the Twenty-Second Conference on Artificial Intelligence (AAAI-07) in Vancouver, July 2007. NICTA is funded by the Australian Government's Department of Communications, Information Technology, and the Arts, and the Australian Research Council. Thanks to Jerome Lang, Michael Maher, Maria Silvia Pini, Steve Prestwich, Francesca Rossi, Brent Venable, and other colleagues for their contributions.

Notes

1. In addition to voting rules that return a single winner, it is interesting to consider extensions such as rules that select multiple winners (for example, to elect a multimember committee) and social welfare functions that return a total ranking over the outcomes.

2. The interested reader is challenged to identify in which 2008 election agents might be voting for Boris, Nick, or Ken.

3. A voting system that returns a total ranking over the outcomes is called a social welfare function.

4. See Conitzer and Sandholm (2006) for the formal definition of weak monotonicity. Informally, monotonicity is the property that improving the vote for an outcome can only help the outcome win.

5. The feasible Pareto optimal outcomes are those outcomes that are feasible and more preferred than any other feasible outcome.

References

Arrow, K. 1970. *Social Choice and Individual Values*. New Haven, CT: Yale University Press.

Bartholdi, J., and Orlin, J. 1991. Single Transferable Vote Resists Strategic Voting. *Social Choice and Welfare* 8(4):341–354.

Bartholdi, J.; Tovey, C.; and Trick, M. 1989a. The Computational Difficulty of Manipulating an Election. *Social Choice and Welfare* 6(3): 227–241.

Bartholdi, J.; Tovey, C.; and Trick, M. 1989b. Voting Schemes for Which It Can Be Difficult to Tell Who Won the Election. *Social Choice and Welfare* 6(2): 157–165.

Black, D. 1948. On the Rationale of Group Decision Making. *Journal of Political Economy* 56(1): 23–34.

Boutilier, C. 2002. A POMDP Formulation of Preference Elicitation Problems. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence,* 239–246. Menlo Park, CA: Association for the Advancement of Artificial Intelligence.

Boutilier, C.; Brafman, R.; Hoos, H.; and Poole, D. 1999. Reasoning with Conditional Ceteris Paribus Preference Statements. In *Proceedings of Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99),* 71–80. San Francisco: Morgan Kaufmann Publishers.

Brafman, R., and Domshlak, C. 2002. Introducing Variable Importance Tradeoffs into CP-nets. In *Proceedings of Eighteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-02),* 69–76. San Francisco: Morgan Kaufmann Publishers.

Brams, S.; Kilgour, D.; and Zwicker, W. 1998. The Paradox of Multiple Elections. *Social Choice and Welfare* 15(2): 211–236.

Conitzer, V. 2007. Eliciting Single-Peaked Preferences Using Comparison Queries. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*. New York: Association for Computing Machinery.

Conitzer, V., and Sandholm, T. 2002. Complexity of Manipulating Elections with Few

Candidates. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence,* 239–246. Menlo Park, CA: Association for the Advancement of Artificial Intelligence.

Conitzer, V., and Sandholm, T. 2003. Universal Voting Protocol Tweaks to Make Manipulation Hard. In *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence,* 781–788. San Francisco: Morgan Kaufmann Publishers.

Conitzer, V., and Sandholm, T. 2006. Nonexistence of Voting Rules That Are Usually Hard to Manipulate. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence,* 239–246. Menlo Park, CA: Association for the Advancement of Artificial Intelligence.

Conitzer, V.; Lang, J.; and Sandholm, T. 2003. How Many Candidates Are Needed to Make Elections Hard to Manipulate. In *Proceedings of the Conference on Theoretical Aspects of Reasoning about Knowledge (TARK IX)*. New York: Association for Computing Machinery.

Conitzer, V.; Sandholm, T.; and Lang, J. 2007. When Are Elections with Few Candidates Hard to Manipulate. *Journal of the Association for Computing Machinery* 54(3).

Domshlak, C.; Prestwich, S.; Rossi, F.; Venable, K.; and Walsh, T. 2006. Hard and Soft Constraints for Reasoning about Qualitative Conditional Preferences. *Journal of Heuristics* 12(4–5): 263–285.

Domshlak, C.; Rossi, F.; Venable, B.; and Walsh, T. 2003. Reasoning about Soft Constraints and Conditional Preferences: Complexity Results and Approximation Techniques. In *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers.

Elkind, E., and Lipmaa, H. 2005. Hybrid Voting Protocols and Hardness of Manipulation. In *Proceedings of the Sixteenth Annual International Symposium on Algorithms and Computation (ISAAC'05)*. Berlin: Springer-Verlag.

Gajdos, T.; Hayashi, T.; Tallon, J.; and Vergnaud, J. 2006. On the Impossibility of Preference Aggregation Under Uncertainty. Working Paper, Centre d'Economie de la Sorbonne, Université Paris 1-Pantheon-Sorbonne.

Gibbard, A. 1973. Manipulation of Voting Schemes: A General Result. *Econometrica* 41(4): 587–601.

Goldsmith, J.; Lang, J.; Truszczynski, M.; and Wilson, N. 2005. The Computational Complexity of Dominance and Consistency in CP-Nets. In *Proceedings of Nineteenth International Joint Conference on Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers.

Hebrard, E.; Hnich, B.; O'Sullivan, B.; and Walsh, T. 2005. Finding Diverse and Similar Solutions in Constraint Programming. In *Proceedings of the Twentieth National Conference on Artificial Intelligence,* 239–246. Menlo Park, CA: Association for the Advancement of Artificial Intelligence.

Hebrard, E.; O'Sullivan, B.; and Walsh, T. 2007. Distance Constraints in Constraint Satisfaction. In *Proceedings of Twentieth International Joint Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.

Konczak, K., and Lang, J. 2005. Voting Procedures with Incomplete Preferences. Paper presented at the IJCAI-2005 Workshop on Advances in Preference Handling, Edinburgh, Scotland, 31 July.

Lacy, D., and Niou, E. 2000. A Problem with Referenda. *Journal of Theoretical Politics* 12(1): 5–31.

Lang, J.; Pini, M.; Rossi, F.; Venable, B.; and Walsh, T. 2007. Winner Determination in Sequential Majority Voting. In *Proceedings of Twentieth International Joint Conference on Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers.

Pini, M.; Rossi, F.; Venable, B.; and Walsh, T. 2007. Incompleteness and Incomparability in Preference Aggregation. In *Proceedings of Twentieth International Joint Conference on Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers.

Prestwich, S.; Rossi, F.; Venable, K.; and Walsh, T. 2005. Constraint-Based Preferential Optimization. In *Proceedings of the Twentieth National Conference on Artificial Intelligence,* 239–246. Menlo Park, CA: Association for the Advancement of Artificial Intelligence.

Procaccia, A. D., and Rosenschein, J. S. 2007. Junta Distributions and the Average-Case Complexity of Manipulating Elections. *Journal of Artificial Intelligence Research* 28: 157–181.

Rossi, F.; Venable, B.; and Walsh, T. 2004. mCP Nets: Representing and Reasoning with Preferences of Multiple Agents. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence,* 239–246. Menlo Park, CA: Association for the Advancement of Artificial Intelligence.

Satterthwaite, M. 1975. Strategy-Proofness and Arrow's Conditions: Existence and Correspondence theorems for Voting Procedures and Social Welfare Functions. *Journal of Economic Theory* 10(2): 187–216.

Walsh, T. 2007a. Manipulating Individual Preferences. Technical Report COMIC-2007-0011, National ICT Australia (NICTA) and the University of New South Wales, Sydney, Australia.

Walsh, T. 2007b. Uncertainty in Preference Elicitation and Aggregation. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence,* 239–246. Menlo Park, CA: Association for the Advancement of Artificial Intelligence.

Xia, L.; Lang, J.; and Ying, M. 2007. Sequential Voting and Multiple Election Paradoxes. In *Proceedings of the Conference on Theoretical Aspects of Reasoning about Knowledge*. New York: Association for Computing Machinery.

Toby Walsh is a senior principal researcher at National ICT Australia (NICTA) in Sydney, conjoint professor at University of New South Wales, external professor at Uppsala University, and an honorary fellow of the School of Informatics at Edinburgh University. He is currently editor-in-chief of the *Journal of Artificial Intelligence Research (JAIR)* and will be program chair of IJCAI-11.