

Preference Handling in Combinatorial Domains: From AI to Social Choice

Yann Chevaleyre
LAMSADE
Univ. Paris-Dauphine

Ulle Endriss
ILLC
University of Amsterdam

Jérôme Lang
IRIT
CNRS & Univ. Paul Sabatier

Nicolas Maudet
LAMSADE
Univ. Paris-Dauphine

Abstract

In both individual and collective decision making, the space of alternatives from which the agent (or the group of agents) has to choose often has a combinatorial (or multi-attribute) structure. We give an introduction to preference handling in combinatorial domains in the context of collective decision making, and show that the considerable body of work on preference representation and elicitation that AI researchers have been working on for several years is particularly relevant. After giving an overview of languages for compact representation of preferences, we discuss problems in voting in combinatorial domains, and then focus on multiagent resource allocation and fair division. These issues belong to a larger field, known as computational social choice, that brings together ideas from AI and social choice theory, to investigate mechanisms for collective decision making from a computational point of view. We conclude by briefly describing some of the other research topics studied in computational social choice.

Introduction

Individual decision making is mostly guided by the agent's preferences over her possible decisions. Similarly, group decision making is guided by the preferences of the agents in the group. For instance, when a group of autonomous agents need to agree on an allocation of resources among themselves, then each individual will judge the outcome according to their own preferences and they will have to transmit parts of these preferences (possibly indirectly and possibly reluctantly so) to their peers in the process of negotiation. Also, to be able to assess whether the negotiation outcome should be considered a "good" allocation (say, whether it reflects a fair agreement) requires knowledge of the individual preferences. Similarly, when voting on a proposition or for a candidate, the ballot submitted by each individual reflects some aspect of their own preferences and the voting protocol in place is charged with aggregating these preferences into a decision that (we hope) constitutes a good reflection of the collective will of the population.

Copyright © 2008, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Collective Decision Making

The classical discipline concerned with the study of mechanisms for collective decision making is *social choice theory* (Arrow, Sen, & Suzumura 2002). Much work in the field has concentrated on normative questions and on establishing abstract results regarding the possibility of designing mechanisms meeting certain requirements. For instance, a seminal result in the field, Arrow's Impossibility Theorem, shows that there can exist no preference aggregation mechanism that would simultaneously satisfy a small number of natural requirements (e.g., the aggregation function should not be dictatorial). Computational concerns, however, have mostly been neglected: What is the computational complexity of the mechanisms proposed by social choice theorists? What are the appropriate algorithmic techniques for these problems? What happens if the number of alternatives to choose from becomes very large?

Such considerations have given rise to an interdisciplinary research effort at the interface of AI and computer science with social choice theory, sometimes dubbed *computational social choice*. On the one hand, computational social choice is concerned with the application of techniques developed in computer science, such as complexity analysis or algorithm design, to the study of social choice mechanisms, such as voting procedures or fair division algorithms. On the other hand, computational social choice seeks to import concepts from social choice theory into AI and computing. For instance, social welfare orderings originally developed to analyze the quality of resource allocations in human society are equally well applicable to problems in multiagent systems or network design.

Combinatorial Domains

Known methods for collective decision making and classical results from social choice theory may not always be applicable when the number of alternatives from which to choose is large. This is particularly true when the set of alternatives has a *combinatorial structure*. Examples include negotiation over indivisible goods (where the number of bundles an agent may obtain is exponential in the number of goods) or the election of a committee (where the number of possible committees is exponential in the number of seats to be filled). For such combinatorial problems, the mere representation of the preferences of individuals over different

alternatives becomes a non-trivial problem. Here methods for preference representation and elicitation developed in AI can make an important contribution.

Below we are going to review some of the languages for compact representation of preferences that are good candidates for modeling problems of collective decision making in combinatorial domains. We are then going to focus on two classes of collective decision making problems where the space of alternatives has a combinatorial structure. The first one is the problem of *voting in combinatorial domains*. Electing a committee rather than a single candidate is a typical example for this problem. The second one is *multiagent resource allocation*, the problem of finding a suitable division of goods among several agents. This is of course a problem that has been studied in multiagent systems and AI for some time. Here we are going to specifically highlight concepts from the social choice and welfare economics literature, which provides useful definitions for what should be considered a fair allocation of goods. We conclude by mentioning some of the other topics that have recently been addressed in the computational social choice literature.

Preferences in Combinatorial Domains

Collective decision making in combinatorial domains first and foremost requires modeling the preferences of individual decision-makers over alternatives with a combinatorial structure. In our discussion of preference representation languages, we start by listing some natural requirements for such languages. Following this general discussion, we are going to describe informally, for the sake of illustration, a couple of concrete languages that have been proposed in the literature, and so-called bidding languages developed for transmitting preferences in the context of combinatorial auctions.

Desirable Properties

When choosing (or designing) a language for representing preferences, there are at least the following five objectives to be considered: (high) expressive power, relative succinctness, (low) complexity, elicitation-friendliness, and cognitive appropriateness. We shall briefly elaborate on each one of these here.

Concerning *expressive power*, a first question to ask is whether the chosen representation language can in fact represent all the preference structures we are interested in. As expressive power will have to be balanced with other considerations, such as complexity and succinctness of representation, the most expressive languages are not always the most attractive ones. If possible, we may want to design our languages so as to be able to exactly express the preference structures of interest, and no others. Also, some languages have more than one way of expressing some particular structures, while others have a unique representation for any preference structure they can express.

If we have two or more languages at our disposal that can all express the preference structures of interest, we may ask which language has more *succinct* representations. We would like to be able to encode a preference structure in as

little space as possible. This point is particularly important for preferences over combinatorial domains. Formally, we say that language \mathcal{L} is at least as succinct as language \mathcal{L}' , with respect to a given class of preference structures if and only if any preference structure from that class that can be expressed in \mathcal{L}' can also be expressed in \mathcal{L} without a significant (that is, super-polynomial) increase in size. Coste-Marquis *et al.* (2004), for example, study the expressive power and relative succinctness of some logic-based languages for representing ordinal preferences.

Third, for any given language, we can analyze the *computational complexity* of various related tasks. Such tasks include, for instance, finding a non-dominated alternative, checking whether an alternative is preferred to another one, whether an alternative is non-dominated, or whether all non-dominated alternatives satisfy a given property. (An alternative is said to be dominated if there is another alternative that is preferred to the former.) Such complexity results can provide hints as to the practical usability of certain languages.

Another criterion to consider is *elicitation-friendliness*: How difficult is it to elicit the preferences of an agent so as to represent them in a given language? This covers both elicitation of preferences from a human user and the design of algorithms for eliciting those parts of a preference structure from a software agent that are actually relevant to the task at hand. The topic of *preference elicitation* is discussed in detail in two other contributions to this special issue (Braziunas & Boutilier 2008; Pu & Chen 2008).

As a final point we mention the issue of *cognitive relevance*. Arguably, preference representation languages that resemble the way humans think about preferences have some advantages, for instance in terms of elicitation from a human subject. By its very nature, this parameter is somewhat more difficult to assess than the other issues mentioned before.

Types of Languages

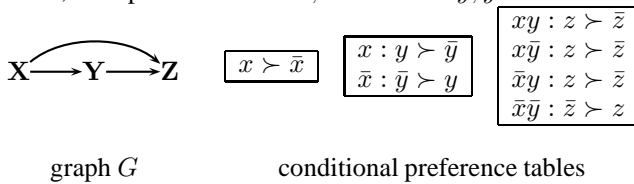
We do not have the space for giving an exhaustive list of languages for modeling preferences in combinatorial domains here, so we attempt only a rough taxonomy and then describe two languages in more detail.

A first dichotomy deals with the nature of the preferences represented: some languages are tailored to compactly represent *cardinal* preferences (utility functions) while others are meant to represent *ordinal* preferences (preference relations). A second dichotomy is concerned with the nature of the language itself. Some of these languages are *graphical*, for instance CP-nets (see below) or GAI-nets (Braziunas & Boutilier 2008): they consist of a structural part that expresses the links between variables, and a “table” part containing the local preferences. Some other languages are based on propositional logic (or possibly a fragment of it): prioritized goals, distance-based goals, weighted goals, bidding languages for combinatorial auctions (see below), and conditional logics of preference. Lang (2004) gives a survey of logical languages for compact preference representation. Finally, some languages are domain-specific (for instance, they may be tailored for expressing bids in auctions), while others are not.

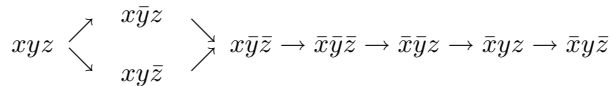
We now describe in more details two typical languages that are commonly used and which are, to some extent, representative of the whole variety of languages. The first is a language for ordinal preferences; the second is a language for cardinal preferences.

CP-nets

Conditional preference networks, or *CP-nets*, are a language for specifying preferences based on the notion of conditional preferential independence (Boutilier *et al.* 2004). Formally, a *CP-net* is a pair consisting of a *directed graph* G whose vertices are the variables and a collection of *conditional preference tables*, one for each variable. The table for variable X contains, for each instantiation of its parent variables in G , a preference relation on the value domain of X . Consider the following CP-net over the two binary variables X and Y , with possible values x, \bar{x} for X and y, \bar{y} for Y :



The directed graph G means that the agent's preference over the values of X is unconditional, and that her preference over the values of Y (resp. Z) is fully determined given the value of X (resp. the values of X and Y). The local preference tables express preferences between the values of a variable, everything else being equal (*ceteris paribus*). For instance, in the table for Y , the item $x : y \succ \bar{y}$ means that when x is true, then y is preferred to \bar{y} for any fixed value of Z . Therefore, in the preference relation expressed by the CP-net, we have $xyz \succ x\bar{y}z$ and $xy\bar{z} \succ x\bar{y}\bar{z}$. The preference relation induced by the CP-net is then the transitive closure of all these preference items directly induced from the conditional preference tables:



In much of the literature, the graph G is assumed to be acyclic. Under this assumption, the induced preference relation is a strict partial order possessing a dominating element.

CP-nets are not fully expressive, because some preference relations are not expressible by CP-nets. On the positive side, preferences expressed as CP-nets are easy to elicit, provided that the graph G is known and possesses a small enough number of edges. To see informally how succinct they are, notice that the space needed to specify a CP-net is exactly the cumulated size of its tables, whereas the explicit representation of the preference relation is always exponentially large. Finally, optimization queries are computationally easy, provided that G is acyclic.

Combinatorial Auction Bidding Languages

Combinatorial auctions are auctions in which the auctioneer is offering not just one but a whole set of goods for sale. Potential buyers can make bids for subsets of this set of goods

and the auctioneer has to choose which of the bids to accept. So-called bidding languages are used to allow agents to communicate bids (in other words, their preferences) to the auctioneer. While strategic considerations may cause agents not to report their *true* preferences, this issue is not relevant from the viewpoint of preference *representation*. Preference structures here are valuation functions mapping bundles of goods to prices. They are usually assumed to be monotonic.

We briefly review the core ideas from the OR/XOR-family of bidding languages (Nisan 2006). Bids are expressed as combinations of atomic bids of the form $\langle S, p \rangle$, where p is the amount the bidder is prepared to pay for the bundle of goods S . In the OR-language, the valuation of a bundle is taken to be the maximal value that can be obtained when computing the sum over *disjoint* bids for subsets of the bundle. For instance, consider the following bid:

$$\langle \{a\}, 2 \rangle \text{ OR } \langle \{b\}, 2 \rangle \text{ OR } \langle \{c\}, 3 \rangle \text{ OR } \langle \{a, b\}, 5 \rangle$$

This expresses that the bidder is willing to pay \$2 for a or b alone, \$3 for c alone, \$5 for both a and b , and \$8 for the full set. The OR-language is not fully expressive as it cannot represent submodular valuations. For example, there is no way to specify that you would only want to pay \$7 for the full set.

In the XOR-language, atomic bids are taken to be mutually exclusive. In this case, the valuation of a bundle is simply the highest value offered for any of its subsets. The XOR-language can express any monotonic valuation function (provided it maps the empty bundle to 0). On the downside, it is typically not very succinct for interesting classes of valuations, as it essentially amounts to enumerating all bundles with non-zero valuation. For instance, the simple valuation function that maps each bundle to its cardinality can be expressed using a linear number of OR-bids, but requires exponential space in the XOR-language. It is also possible to combine the OR and the XOR operator to obtain bidding languages with better expressiveness and succinctness properties than the pure OR- and XOR-languages.

An interesting alternative is to simulate XOR-bids by means of OR-bids. The idea is to introduce so-called *dummy items*, which have no function other than making bundles mutually exclusive. For instance, to express that the set $\{a, b, c\}$ in our earlier example should be valued at \$7, we could add the dummy item d to obtain $\langle \{c, d\}, 3 \rangle$ and $\langle \{a, b, d\}, 5 \rangle$, and to bid in addition on $\langle \{a, b, c\}, 7 \rangle$. This bidding language, known as the OR*-language, is as expressive as the XOR-language.

Another approach to designing bidding languages has been to use propositional formulas describing desirable combinations of goods as bids and to allow arbitrary subformulas of these formulas to be annotated with prices (Boutilier & Hoos 2001). Such languages belong to a more general family of languages that make use of formulas of propositional logic to characterize desirable aspects of the alternatives to be decided upon and to pair these formulas with numerical weights indicating their importance (see for instance (Chevalere, Endriss, & Lang 2006) for expressivity and succinctness results regarding such languages).

Voting in Combinatorial Domains

In many contexts, a group of voters has to make a common decision on several possibly related issues. For instance, imagine a set of friends have to agree on a common menu to be composed of a first course, a main course, a dessert, and a wine. Some of them may have preferential dependencies, for instance, they may prefer white wine if the main course is fish, and red wine otherwise. Another example would be that the inhabitants of some local community have to make a joint decision over several related issues of local interest, for instance, to decide whether some new public facility such as a swimming pool or a tennis court should be built. Such elections are called *multiple referenda*. A third example are committee elections. Suppose the members of an association have to elect a steering committee, composed of a president, a vice-president and a treasurer. In such situations, voters typically have preferential dependencies, for instance they would not like the president and the treasurer to be close friends (nor enemies).

Multiple Election Paradoxes

As soon as voters have preferential dependencies between issues, it is generally a bad idea to decompose a vote problem on p issues into a set of p smaller problems, each one bearing on a single issue, because this can give rise to “multiple election paradoxes”. Such paradoxes have been studied by a number of authors (Brams, Kilgour, & Zwicker 1998; Benoit & Kornhauser 1999; Lacy & Niou 2000).

Consider the following example. A joint decision has to be made about whether or not to build a new swimming pool (S or \bar{S}) and a new tennis court (T or \bar{T}). Assume that the preferences of voters 1 and 2 are $S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$, those of voters 3 and 4 are $\bar{S}T \succ ST \succ \bar{S}\bar{T} \succ ST$, and those of voter 5 are $ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$.

The first complication in this example is that voters 1 to 4 will feel ill at ease when asked to report their projected preference on $\{S, \bar{S}\}$ and $\{T, \bar{T}\}$. Only voter 5 knows that whatever the other voters’ preferences about $\{S, \bar{S}\}$ (resp. $\{T, \bar{T}\}$), she can vote for T (resp. S) without any risk of experiencing regret. Experimental studies suggest that most voters tend to report their preferences optimistically in such situations; for instance, voters 1 and 2 would likely report a preference for S over \bar{S} .

The second problem (the paradox itself) is that under this assumption that voters report optimistic preferences, the outcome will be ST , which is *the worst outcome for all but one voter*. How can such paradoxes be avoided? Reformulating the question in a more constructive way: how should a vote on related issues be conducted?

Possible Solutions

We can list five ways of proceeding, each of which has its own pitfalls. The first four work at the global level, where voters vote for combinations of values, while the last one works at the local level (via a decomposition of the problem).

Solution 1 is to ask voters to report their entire preference relation explicitly on the set of alternatives, and then apply any fixed voting rule.

Solution 2 consists in asking voters to report only a small part of their preference relation (for instance, their their k most preferred outcomes, where k is a small number) and apply a voting rule that needs this information only. The plurality rule, for instance, which chooses the candidate ranked first by the highest number of voters, is such a rule.

Solution 3 consists in limiting the number of possible combinations that voters may vote for.

Solution 4 requires each voter to express her preferences as an input in some fixed compact representation language, and then applies a fixed voting rule to the profile consisting of the preference relations induced by the voters’ inputs.

Finally, Solution 5 imposes a partitioning of the domain and requires the voters to vote separately on each issue, either simultaneously or sequentially (in the latter case, the outcome of the vote on one issue is revealed to the voters before they vote on subsequent issues).

Let us now analyze these five proposals. Solution 1 works only if the number of issues is very small. Suppose we have 10 binary issues. Then voters surely would not want to bother spending a few hours (and lots of energy) enumerating 2^{10} alternatives! Solution 2 requires little communication, but this is its only merit, as it is likely to give catastrophic results as soon as the number the issues is not very small. For instance, using the plurality rule when the number of issues is significant and the number of voters is small could well result in a situation where no outcome gets more than one vote, in which case plurality would give an extremely poor (namely entirely random) result. Solution 3, advocated by Brams *et al.* (1998), presents the chairperson with a very problematic choice and introduces a strong level of arbitrariness. Solution 4, also known as “combinatorial vote” (Lang 2004), presents two difficulties: first, the language has to be elicitation-friendly; and second, the complexity of computing the outcome is very high in most cases.

Solution 5 has been first supported by Lacy and Niou (2000), who show that if the voters’ preferences are separable (which means that each voter’s preference on the values of an issue is independent from the outcome on other issues), then the approach is safe. However, separability is a very demanding assumption that is unlikely to be met in practice. Several recent papers (Lang 2007; Xia, Lang, & Ying 2007) impose a much weaker domain restriction than separability under which sequential voting can be applied “safely”: informally, the condition should be that each time a voter is asked to report his preferences on a single issues or a small set of issues, these preferences do not depend on the values of the issues that have not been decided yet. Formally, this can be expressed as the following condition: there is a linear order $\mathcal{O} = \mathbf{X}_1 > \dots > \mathbf{X}_p$ on the set of issues such that for every voter v and every j , the preferences of v on \mathbf{X}_j are preferentially independent from $\mathbf{X}_{j+1}, \dots, \mathbf{X}_p$ given $\mathbf{X}_1, \dots, \mathbf{X}_{j-1}$. If this property is satisfied, then *sequential voting rules* can be defined in the following way. Let r_1, \dots, r_n be voting rules on the domains of $\mathbf{X}_1, \dots, \mathbf{X}_n$ respectively. Then the *sequential composition* of r_1, \dots, r_n is defined as follows: first, since all voters have unconditional preferences over the values of \mathbf{X}_1 , r_1 can be applied to these preferences about \mathbf{X}_1 so as to make a de-

cision on the value of X_1 ; then, given this value d_1 chosen for X_1 , voters have well-defined preferences over the values of X_2 that are independent of X_3, \dots, X_n , therefore r_2 is applied to these local preferences over X_2 , and so on. In order to compute the outcome of these sequential voting rules we do not need to know the voters' full preference relations: it suffices for each voter to express a CP-net, with the condition that the dependency graph of the CP-nets is *acyclic* and *common to all voters*.

As an example, consider two binary variables X and Y , with the ordering $X > Y$. Each local rule is the plurality rule (which, since domains are binary, coincides with the majority rule). Consider seven voters, three of whom express the preference relation $\bar{x}y \succ \bar{x}\bar{y} \succ x\bar{y} \succ xy$, two $xy \succ x\bar{y} \succ \bar{x}\bar{y} \succ \bar{x}y$ and two $x\bar{y} \succ xy \succ \bar{x}y \succ \bar{x}\bar{y}$. The voters' preferences over the values of X are first elicited (which is easy, since, by assumption, these preferences are unconditional). Since four voters out of seven prefer x to \bar{x} , the decision $X = x$ is taken. Now, voters are asked about their preferences on the values of Y given $X = x$. Given $X = x$, five voters out of seven prefer \bar{y} to y , therefore the decision $Y = \bar{y}$ is taken and the final decision is $x\bar{y}$.

Preference Aggregation

So far we have focused on *voting*. While a voting rule outputs a single candidate, an *aggregation function* outputs a collective preference relation or utility function. Ideally, if the voters' preferences are represented in some language, we would like the output to be represented in the same language: for instance, we would want to aggregate a collection of GAI-nets into a GAI-net (Gonzales, Perny, & Queiroz 2006), or similarly for CP-nets (Rossi, Venable, & Walsh 2004).

Multiagent Resource Allocation and Fair Division

Another important application domain that has attracted a great deal of attention in computer science and AI in recent years is multiagent resource allocation (Chevalerey *et al.* 2006). Different variants have been studied, but a typical setting involves a set of indivisible goods that need to be distributed among a set of agents. Typically, goods exhibit different kinds of synergies between them and cannot be considered independently without risking undesirable outcomes. This means that agents have to express preferences over a combinatorial domain. Specifically, the number of bundles of goods an agent might receive is exponential in the number of goods. The allocation mechanism that has been studied most is that of *combinatorial auctions*, where the efficiency criterion used to evaluate the quality of an allocation is typically the revenue of the auctioneer.

Multiagent resource allocation shares some obvious similarities with voting in combinatorial domains: agents report preferences on several alternatives (here, possible allocations of goods to agents), and a collective decision has to be taken on the allocation that should be selected as the outcome of the procedure. Conitzer (2008) discusses some of these similarities between voting and resource allocation

(focusing on combinatorial auctions, as one class of allocation procedures) in detail. It is useful though, to also emphasize some of the distinctive features of resource allocation problems, as opposed to voting. Among these are the lack of so-called *externalities*, the fact that *payments* may enter the process, and the consideration of *fairness* requirements.

Regarding the first of these, in the context of resource allocation problems, agents are mostly assumed to only worry about *their own* bundle of resources, and to be indifferent to what the others receive. This is known as the "no externalities" assumption. In other words, in the context of an election, all voters are asked to express a preference over the same set of alternatives, while in multiagent resource allocation they only express preferences regarding their own lot. Solving a resource allocation problem by having agents vote on possible allocations and then applying a voting rule would be conceivable in theory, but it would also be unnecessarily complex.

A second distinctive feature is that allocation problems often include a monetary component that is not found in voting. Including the possibility for monetary payments between agents enlarges the range of possible deals that can be made and, using techniques from mechanism design, such payments can also be used to give agents incentives to truthfully reveal their preferences.

As mentioned above, a third distinctive feature of resource allocation is the consideration of fairness issues. More generally speaking, there is a whole range of different (economic) efficiency and fairness criteria that we may wish to apply to assess the quality of an allocation of resources to agents. The distinction between fairness and efficiency criteria is best illustrated with an example.

Fairness and Efficiency

The allocation of Earth observation satellite images provides a real-world illustrative example where indivisible goods have to be allocated to a group of agents. The cost of space projects is so high that they usually need to be co-funded by several agents (countries or large companies). When the satellite is eventually in space circling the Earth, we are faced with the problem of how to allocate images to the parties who funded the project. Each agent can request any number of images to be taken, but for technical reasons not all of these requests can be satisfied. How should we decide which photos to take?

On the one hand, *efficiency* considerations suggest that we should aim at maximizing the overall number of pictures taken. On the other hand, *fairness* requirements may dictate that we cannot always disappoint the same agent, even if this reduces the total number of requests that can be satisfied. For instance, we may wish to ensure that, over time, each agent receives a return on investments that is at least roughly proportional to their financial contribution. Social choice theory and welfare economics (Arrow, Sen, & Suzumura 2002) have produced a host of mathematically precise criteria to formally assess efficiency and fairness. Some of these have been exploited in AI for some time, while others are less well-known in this community.

Possibly the best known efficiency criterion is *Pareto optimality*: it stipulates that the chosen agreement should be such that no alternative agreement would be better for some agents without being worse for any of the others. A stronger requirement would be to ask for an allocation with maximal *utilitarian social welfare*, the sum of utilities of the individual agents. The classical counterpart to utilitarianism is egalitarianism: maximizing *egalitarian social welfare* requires choosing an allocation that would maximize the utility of the agent worst off. A refinement of this concept is the so-called *leximin-ordering*: here we first try to maximize the well-being of the poorest agent; once this option has been exhausted we turn to maximizing the well-being of the second poorest agent, and so forth. Another interesting fairness criterion is *envy-freeness*: if possible, we would like to find an allocation such that no agent would rather have a bundle that has been allocated to any of its peers. Envy-free allocations do not always exist, in which case we may, for instance, choose to aim at minimizing the number of envious agents or the degree of envy experienced by any one agent.

One type of problem that has been investigated in recent work is the computational hardness of finding an allocation of resources that is optimal according to one of these social criteria. Typically, these problems are difficult. Sometimes, but not always, putting severe restrictions on the range of preference structures can offer an escape route though. For instance, finding an allocation that maximizes utilitarian social welfare is known to be NP-hard in general (it corresponds to the SET-PACKING problem), but it becomes easy if all agents have additively separable preferences. On the other hand, maximizing egalitarian social welfare remains NP-hard even in this seemingly simple case. Questions related to envy-freeness give rise to particularly interesting computational questions (Lipton *et al.* 2004; Bouveret & Lang 2008).

Centralized vs. Distributed Allocation

We can distinguish different types of allocation procedures, depending on whether they are centralized or distributed. An example for the former are combinatorial auctions. Here, a single auctioneer is in charge of the “winner determination problem”: determining the allocation of goods and fixing the prices at which the bundles should be sold. Solving this problem is equivalent to finding an allocation with maximal utilitarian social welfare, which we have seen before to be NP-hard in the general case. Still, in recent years several algorithms have been developed that perform well in practice. Also, for many real-world applications, preferences structures often exhibit some regularities that make the problem tractable even in theory (Lehmann, Müller, & Sandholm 2006).

A different perspective on the same application problem can be taken if we assume that no central agent is dedicated to the elicitation of the others’ preferences and to the computation of an optimal outcome. Under that view, agents negotiate locally, by accepting or rejecting deals proposed by some other agents (with or without side payments), until a stable situation is reached (when no more deals are possible). A possible criterion to apply what deals should be

considered acceptable is for each agent to insist on an *immediate* benefit to themselves (such agents are said to be myopic). Additionally, we may also put structural restrictions on deals (how many agents, or goods, can be involved in a given exchange). Such restrictions put severe limitations on what can be achieved in a negotiation system.

For instance, under the assumption of myopic rationality, an agent holding two resources and valuing this bundle as “useful” would never give one of them to another agent valuing the same bundle as “extremely useful” if single resources are completely useless to both of them. If the system only allows deals involving a single resource at a time, it could remain stuck in such a local optimum. On the other hand, if no structural restrictions are present, the negotiation among such rational agents is known to always converge to an allocation with maximal utilitarian social welfare, if side payments are allowed (Sandholm 1998).

An important question is then to characterize, for a given restriction on possible deals, what restrictions on preference structures still allow us to guarantee convergence of the system to a global optimum. This question may be asked for a range of different social optimality criteria, such as maximal utilitarian/egalitarian social welfare, Pareto optimality, or envy-freeness. In general, when nothing is assumed regarding the agents’ preference structures, the highest structural complexity of deals is required (if convergence is possible at all), which is of little practical interest of course. Unfortunately, some basic natural restrictions, such as monotonicity, tend not to help: it may still be possible that some specific negotiation scenario would require a very complex deal to reach an optimal allocation. A challenging question is then to identify the class of preference structures that fits a given protocol restriction, in the sense that it still allows a guarantee of this convergence property. In other words: given a class of deals allowed under some negotiation protocol and a suitable efficiency or fairness criterion for assessing the quality of allocations, can we identify conditions on the preferences of the negotiating agents that would guarantee that any sequence of acceptable deals would be bound to converge to an allocation that is optimal under the chosen criterion? Endriss *et al.* (2006) discuss the problem of convergence in distributed resource allocation in detail.

Conclusion

Preference handling in combinatorial domains and its application to voting and fair division problems are examples for ongoing research efforts in the field of *computational social choice*. To conclude, we briefly mention some of the other research directions that have recently been explored. As this is a very active area with a fast growing body of literature, we make no attempt at being comprehensive and we are only going to cite a handful of exemplary contributions.

Much work in computational social choice applies the tools of computational complexity theory to social choice settings, particularly to the analysis of voting rules. For example, while it is computationally easy to compute the winner for most of the voting rules in regular use, there are also rules for which this problem turns out to be computationally intractable. An example is the rule proposed

by C.L. Dodgson (of “Alice in Wonderland” fame) in 1876: elect the candidate who is closest to being able to beat each other candidate in a pairwise competition, for a suitable definition of “closest” (Bartholdi, Tovey, & Trick 1989b; Hemaspaandra, Hemaspaandra, & Rothe 1997).

Computational complexity has also been suggested as a barrier against manipulation in elections. Classical results from social choice theory show that for any voting rule to choose between three or more candidates that is not dictatorial (meaning that one voter always determines the outcome), there will be situations in which some voters may have an incentive to manipulate, in the sense of submitting a ballot that does not truthfully reflect their real preferences. For instance, if you have an inkling that your favorite candidate has no chance of winning, you may be tempted to vote for your second-best choice instead. In an ideal world, this kind of reasoning should not be necessary. So one direction of research that is currently being pursued is to search for voting procedures that make it computationally hard to manipulate in this manner. While there are several results establishing NP-hardness, beginning with the seminal work of Bartholdi *et al.* (1989a), the most recent discussion has concentrated on the question to what extent such worst-case complexity results offer sufficient protection in practice, and whether or not meaningful average-case complexity results are achievable (Conitzer & Sandholm 2006; Procaccia & Rosenschein 2007). Besides manipulation in the aforementioned sense, the complexity of other forms of election control, e.g., by strategically entering additional candidates into the race, has also been studied (Faliszewski *et al.* 2008).

Besides computational complexity, also the theory of communication complexity has been applied to social choice settings (Conitzer & Sandholm 2005). How much information needs to be exchanged to determine the winner of an election or to support a particular negotiation protocol is of course especially relevant in combinatorial domains. But also in other settings it is interesting to try to quantify the degree of privacy that can be afforded to the individuals taking part in a collective decision making process.

There have also been computational studies of so-called cake-cutting procedures (Sgall & Woeginger 2007). Designing a protocol for agents to divide a cake (a single divisible good) between them is the canonical example for a fair division problem. For instance, if we interpret fairness as envy-freeness, then the problem of fairly dividing a cake between more than three agents such that each player receives a connected piece (that is, not a collection several small pieces from different parts of the cake) is still an open problem. Observe that, contrary to what we have discussed in this paper, cake-cutting is not a combinatorial problem.

Yet another line of work aims at developing logics for modeling social choice procedures. Just as computer scientists have long been using logic to formally specify the behavior of computer systems, to allow automatic verification of certain desirable properties of such systems, it appears promising to develop suitable logics that would allow the formal specification of social choice procedures. This line of research is also known as *social software* (Parikh 2002).

For a somewhat more technical introduction to computational social choice we refer the reader to our recent survey paper on the topic (Chevalerey *et al.* 2007), which cannot claim to be complete either, but which does reference a large number of works for further reading. In a recent *AI Magazine* article, Walsh (2007) also reviews several branches of computational social choice (including manipulation, elicitation and uncertainty) that we did not discuss in detail here. Finally, another rich source of information are the proceedings of the 1st and 2nd International Workshops on Computational Social Choice (COMSOC-2006 and COMSOC-2008).

Acknowledgments. The work of the second author has been partially supported by the Dutch NWO Vidi project “Collective Decision Making in Combinatorial Domains”. The work of the other three authors has been partially supported by the French ANR project “Preference Handling and Aggregation in Combinatorial Domains”. We would like to thank the reviewers of this paper for their helpful feedback.

References

- Arrow, K. J.; Sen, A. K.; and Suzumura, K., eds. 2002. *Handbook of Social Choice and Welfare*. North-Holland.
- Bartholdi, J. J.; Tovey, C. A.; and Trick, M. A. 1989a. The computational difficulty of manipulating an election. *Social Choice and Welfare* 6(3):227–241.
- Bartholdi, J. J.; Tovey, C. A.; and Trick, M. A. 1989b. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare* 6(3):157–165.
- Benoit, J.-P., and Kornhauser, L. 1999. On the assumption of separable assembly preferences. *Social Choice and Welfare* 16(3):429–439.
- Boutilier, C., and Hoos, H. H. 2001. Bidding languages for combinatorial auctions. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI-2001)*. Morgan Kaufmann.
- Boutilier, C.; Brafman, R. I.; Domshlak, C.; Hoos, H. H.; and Poole, D. 2004. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research* 21:135–191.
- Bouveret, S., and Lang, J. 2008. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *Journal of Artificial Intelligence Research*. To appear.
- Brams, S.; Kilgour, D.; and Zwicker, W. 1998. The paradox of multiple elections. *Social Choice and Welfare* 15(2):211–236.
- Braziunas, D., and Boutilier, C. 2008. Elicitation of factored utilities. *AI Magazine*. In this issue.
- Chevalerey, Y.; Dunne, P. E.; Endriss, U.; Lang, J.; Lemaître, M.; Maudet, N.; Padget, J.; Phelps, S.; Rodríguez-Aguilar, J. A.; and Sousa, P. 2006. Issues in multiagent resource allocation. *Informatica* 30:3–31.

- Chevalere, Y.; Endriss, U.; Lang, J.; and Maudet, N. 2007. A short introduction to computational social choice. In *Proc. 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-2007)*. Springer-Verlag.
- Chevalere, Y.; Endriss, U.; and Lang, J. 2006. Expressive power of weighted propositional formulas for cardinal preference modelling. In *Proc. 10th International Conference on Principles of Knowledge Representation and Reasoning (KR-2006)*, 145–152. AAAI Press.
- Conitzer, V., and Sandholm, T. W. 2005. Communication complexity of common voting rules. In *Proc. 6th ACM Conference on Electronic Commerce (EC-2005)*. ACM Press.
- Conitzer, V., and Sandholm, T. W. 2006. Nonexistence of voting rules that are usually hard to manipulate. In *Proc. 21st National Conference on Artificial Intelligence (AAAI-2006)*, 627–634. AAAI Press.
- Conitzer, V. 2008. Comparing multiagent systems research in combinatorial auctions and voting. In *Proc. 10th International Symposium on Artificial Intelligence and Mathematics (ISAIA-2008)*.
- Coste-Marquis, S.; Lang, J.; Liberatore, P.; and Marquis, P. 2004. Expressive power and succinctness of propositional languages for preference representation. In *Proc. 9th International Conference on Principles of Knowledge Representation and Reasoning (KR-2004)*. AAAI Press.
- Endriss, U.; Maudet, N.; Sadri, F.; and Toni, F. 2006. Negotiating socially optimal allocations of resources. *Journal of Artificial Intelligence Research* 25:315–348.
- Faliszewski, P.; Hemaspaandra, E.; Hemaspaandra, L.; ; and the, J. R. 2008. A richer understanding of the complexity of election systems. In Ravi, S., and Shukla, S., eds., *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*. Springer-Verlag. To appear.
- Gonzales, C.; Perny, P.; and Queiroz, S. 2006. Preference aggregation in combinatorial domains using GAI-networks. In *DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, 165–179.
- Hemaspaandra, E.; Hemaspaandra, L. A.; and Rothe, J. 1997. Exact analysis of Dodgson elections: Lewis Carroll’s 1876 voting system is complete for parallel access to NP. *Journal of the ACM* 44(6):806–825.
- Lacy, D., and Niou, E. 2000. A problem with referendums. *Journal of Theoretical Politics* 12(1):5–31.
- Lang, J. 2004. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence* 42(1–3):37–71.
- Lang, J. 2007. Voting and aggregation on combinatorial domains with structured preferences. In *Proc. 20th Joint International Conference on Artificial Intelligence (IJCAI-2007)*.
- Lehmann, D.; Müller, R.; and Sandholm, T. W. 2006. The winner determination problem. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press.
- Lipton, R. J.; Markakis, E.; Mossel, E.; and Saberi, A. 2004. On approximately fair allocations of indivisible goods. In *Proc. 5th ACM Conference on Electronic Commerce (EC-2004)*. ACM Press.
- Nisan, N. 2006. Bidding languages for combinatorial auctions. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press.
- Parikh, R. 2002. Social software. *Synthese* 132(3):187–211.
- 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.
- Pu, P., and Chen, L. 2008. User-involved preference elicitation and revision. *AI Magazine*. In this issue.
- Rossi, F.; Venable, K.; and Walsh, T. 2004. mCP nets: Representing and reasoning with preferences of multiple agents. In *Proc. 19th National Conference on Artificial Intelligence (AAAI-2004)*, 729–734. AAAI Press.
- Sandholm, T. W. 1998. Contract types for satisficing task allocation: I Theoretical results. In *Proc. AAAI Spring Symposium: Satisficing Models*.
- Sgall, J., and Woeginger, G. J. 2007. An approximation scheme for cake division with a linear number of cuts. *Combinatorica* 27(2):205–211.
- Walsh, T. 2007. Representing and reasoning with preferences. *AI Magazine* 28(4):59–70.
- Xia, L.; Lang, J.; and Ying, M. 2007. Sequential voting and multiple election paradoxes. In *Proc. 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2007)*.