

Comparing Multiagent Systems Research in Combinatorial Auctions and Voting*

Vincent Conitzer

Department of Computer Science
Duke University
Durham, NC 27708, USA
conitzer@cs.duke.edu

Abstract

In a combinatorial auction, a set of resources is for sale, and agents can bid on subsets of these resources. In a voting setting, the agents decide among a set of alternatives by having each agent rank all the alternatives. Many of the key research issues in these two domains are similar. The aim of this paper is to give a convenient side-by-side comparison that will clarify the relation between the domains, and serve as a guide to future research.

Introduction

In multiagent systems, it is often necessary for a group of agents to make a collective decision even though they have different preferences over the different options. For example, the agents may have to decide how to allocate a set resources among themselves. A common mechanism for doing this is to run a *combinatorial auction*, where the agents place bids on bundles (subsets) of resources (for example, an agent can bid 10 on the bundle of resources consisting of A and C), and based on these bids an allocation of resources is determined, as well as the payment that each agent needs to make. However, there are many other collective decision problems that do not involve allocating resources or making payments. A general approach for choosing among a set of alternatives is for each agent to rank all the alternatives, after which a winning alternative is chosen based on these rankings. In this case, we say that the agents *vote* over the alternatives (and the rankings are the *votes*). Voting settings are also referred to as *social choice* settings.

Combinatorial auctions have become a well-established research topic in multiagent systems. In recent years, research on voting in multiagent systems has also soared, and a community of people interested in *computational social choice* has formed. While there is certainly significant interaction between the people working on combinatorial auctions and the ones working on voting, the two communities are perhaps more disjoint than one would expect. This paper aims to compare some of the research issues across the two topics, thereby providing to each community a convenient window into the other (and, perhaps, a window into both for outsiders). The comparison is also likely to suggest new research directions for each topic. Finally, we argue that the

research on the two topics is likely to converge further in the future.

Defining the rules

Both in combinatorial auctions and in voting settings, we need to specify how agents report their preferences (that is, how they place their bids/cast their votes), and how an outcome is chosen on the basis of these. As we will see, there are multiple ways of doing so in each case.

Combinatorial auctions

To understand the different ways in which a combinatorial auction can be designed, it is helpful to first study some common single-item auctions.

- *English auction.* In an English auction (perhaps the best-known auction format) any bidder can enter a bid higher than the current highest bid at any point. Once nobody wants to submit a higher bid, the current highest bidder wins the item and pays her bid.
- *Japanese auction.* In a Japanese auction, there is an initial price of zero for the item, which is gradually increased. A bidder can leave the room at any point if the price becomes too high for her. The auction ends when only one bidder remains, who then wins and pays the final price.
- *Dutch auction.* In a Dutch auction, the price starts at a high value and is gradually *decreased*. At any point, any bidder can claim the item at the current price, at which point the auction ends.
- *First-price sealed-bid auction.* Each bidder privately sends a bid to the auctioneer (for example, in a sealed envelope). The highest bidder wins and pays her bid.
- *Second-price sealed-bid auction.* This auction is identical to the first-price sealed-bid auction, except the highest bidder (who still wins) pays the bid of the *second*-highest bidder.

A common objective among all of these auctions is to allocate the item to the bidder who values the item the most (that is, to allocate the item *efficiently*). The differences among them mostly reflect other issues, such as the following. One advantage of the English, Japanese, and Dutch auctions is that, depending on what happens in the auction, a

*Copyright retained by the author.

bidder may not have to invest the effort necessary to determine her exact valuation for the item. For example, in an English or Japanese auction, once the second-to-last bidder drops out, the winner no longer has to think about how much longer she would have stayed in the auction; and in a Dutch auction, once someone else claims the item, the remaining bidders no longer have to think about when they would have claimed the item. This is closely related to *preference elicitation*, which we will discuss in a later section. An advantage of the second-price sealed-bid auction over the first-price sealed-bid auction is the following. In the first-price auction, bidders will try to only slightly outbid their competitors, to pay as little as possible. If the bidder with the highest value underestimates the other bids, this can result in another bidder winning the item, so that the allocation is inefficient. In the second-price sealed-bid auction, however, there is no reason to try to only slightly outbid the next bidder; in fact, it can be shown that it is optimal to bid one's true valuation for the item. We will discuss issues such as these in a later section.

The main point, however, is that the various auctions agree, in some sense, on the objective of efficient allocation, and the differences merely reflect other issues. There are other auctions that have a different objective, such as revenue-maximizing auctions (Myerson 1981), which do not always allocate the item efficiently; or even auctions that try to minimize revenue (although these still allocate the item efficiently) (Bailey 1997; Porter, Shoham, & Tennenholtz 2004; Cavallo 2006; Guo & Conitzer 2007; Moulin 2007). Nevertheless, in combinatorial auctions, typically the main objective is to allocate efficiently (though there is some work on revenue-maximizing combinatorial auctions as well (Avery & Hendershott 2000; Armstrong 2000; Conitzer & Sandholm 2004; Likhodedov & Sandholm 2004; 2005)). That is, it is assumed that every bidder i has a valuation function $v_i : 2^I \rightarrow \mathbb{R}$, where $v_i(S)$ is bidder i 's value for the subset S of the items I , and the goal is to allocate to the bidders nonoverlapping bundles S_i to maximize $\sum_i v_i(S_i)$. As in the single-item auction case, there are distinctions among combinatorial auctions in terms of the temporal aspects of the auction as well as the payments to be made by the bidders, but these distinctions are again driven by other considerations, which we will discuss in later sections. (There are also variants of combinatorial auctions, such as combinatorial *reverse auctions*, where the auctioneer seeks to *buy* certain items and the bidders offer to *sell* bundles of these items at various prices; and combinatorial *exchanges*, where agents can be both buyers and sellers (Sandholm *et al.* 2002). While these variants are very important, they resemble regular combinatorial (forward) auctions in terms of the issues discussed in this paper, so we will not consider them in the remainder.)

Voting

In a typical voting setting, there is a set of *alternatives* or *candidates*, C , and each agent (voter) i has preferences \succeq_i over these m alternatives, with $a \succeq_i b$ indicating that i (weakly) prefers a over b . Usually, for convenience, it is assumed that all preferences are strict, so that we simply use

\succ_i . (Recent work has begun to extend social choice theory to settings where there is *incompleteness/incomparability* in the agents' preferences (Pini *et al.* 2005; Rossi *et al.* 2006; Pini *et al.* 2007).) Based on the preferences that the voters report, one alternative is chosen as the winner. (Sometimes, the output is a complete (aggregate) ranking of the alternatives; generally any rule can be used to determine either just a winner or an aggregate ranking.) Some example rules are given below. (For each rule that gives points to alternatives, the alternative with the highest score is the winner.)

- *Plurality*. An alternative receives a point every time it is ranked first.
- *Borda*. An alternative receives $m - 1$ points every time it is ranked first, $m - 2$ every time it is ranked second, ..., and 0 every time it is ranked last.
- *Copeland*. We conduct a *pairwise election* between every pair of alternatives: the winner of the pairwise election is the alternative that is ranked higher by more voters. An alternative receives 2 points for every pairwise win, 1 point for every pairwise tie, and 0 points for every pairwise loss.
- *Bucklin*. If there is an alternative that is ranked first by more than half the voters, that alternative wins; otherwise, if there is an alternative that is ranked first or second by more than half the voters, that alternative wins; *etc.*
- *STV*. The alternative that is ranked first the *fewest* times is removed from every vote. (Votes that had this alternative ranked first will now have another alternative ranked first.) This is repeated until one alternative remains.
- *Slater*. We choose an aggregate ranking of the alternatives that is consistent with the outcome of as many pairwise elections as possible.
- *Kemeny*. We choose an aggregate ranking of the alternatives that has as few disagreements with votes as possible (a *disagreement* occurs when a vote ranks a above b , but the aggregate ranking ranks b above a).

In some sense, the differences among these rules are more fundamental than the differences among (combinatorial) auctions. Most auctions agree on the objective of efficient allocation, and the differences among them are due more to other considerations, such as incremental preference revelation and strategic bidding. In contrast, in a voting setting, it is not even clear what objective we should be pursuing. Certainly we can state a vague objective such as "the winner should be high in the preferences of many voters and low in the preferences of few," but there are many different interpretations of this, and, in some sense, each of the voting rules above corresponds to one such interpretation.

One natural approach to identifying the "optimal" voting rule is to specify some axioms that such a rule should satisfy. Unfortunately, there are several impossibility results that show that *no* rule satisfies certain natural properties. For example, consider the *independence of irrelevant alternatives* (IIA) criterion, which states that, if we modify the votes but do not change whether a is ahead of b in any vote, then in the aggregate ranking, there also should be

no change in whether a is ahead of b . While this is a natural criterion, *Arrow's impossibility theorem* (Arrow 1963) states that with 3 or more alternatives, any rule that satisfies IIA must either be *Pareto inefficient*, which means that it sometimes ranks b ahead of a even though all voters rank a ahead of b , or *dictatorial*, which means that the rule simply copies a fixed voter's ranking. There are some combinations of other axioms that are satisfied by exactly one rule—for example, Young and Levenglick (Young & Levenglick 1978) give an axiomatic characterization of the Kemeny rule. However, such axiomatic arguments have so far not succeeded in building broad consensus on what the optimal rule is.

Another approach that has been pursued to decide on the optimal voting rule is the following. Suppose that there exists a “correct” outcome (winner or ranking), which we cannot directly observe; but every voter's preferences constitute a noisy observation of this correct outcome. Given a noise model (a conditional probability distribution over the preferences given the correct outcome), it would make sense to choose the outcome that maximizes the likelihood of the observed preferences. This approach was already pursued by the early social choice theorist Condorcet (de Caritat (Marquis de Condorcet) 1785), who proposed one particular noise model. In this noise model, each voter ranks each pair of alternatives correctly with some probability $p > 1/2$, independently. (This can lead to cyclic preferences, but that does not affect the maximum likelihood approach.) Condorcet solved for the maximum likelihood estimator rule for the cases of 2 and 3 alternatives. Two centuries later, in a paper titled “Optimal Voting Rules” (Young 1995), Young showed that the solution for general numbers of alternatives coincides with the Kemeny rule. Unfortunately, this argument is convincing only to the extent that one believes that Condorcet's noise model is the correct one. In fact, we showed more recently that for many (but not all) of the common voting rules, there is a noise model such that that rule becomes the solution (Conitzer & Sandholm 2005a).

It thus appears that for the foreseeable future, no general agreement will emerge on which rule is optimal. Moreover, as we move towards settings in which there are so many alternatives that it is no longer feasible for a voter to provide a full ranking of all of them, new rules must be designed to address this. We will discuss this in more detail in the next section.

Expressing preferences

In the previous section, we defined the structure of the agents' preferences (valuations for bundles in combinatorial auctions, and rankings of alternatives in voting). In order to choose an outcome based on the agents' preferences, the agents will need to report them; and to do so, the agents need a *language* in which to express their preferences. It is generally easy to create a straightforward language for doing so, but expressing one's preferences in such a language often requires exponential space. A good language for expressing preferences will allow agents to specify “natural” preferences concisely.

Combinatorial auctions

In a combinatorial auction, the straightforward way for a bidder to communicate her preferences to the auctioneer is to simply provide a list of $2^{|I|} - 1$ values, one for each nonempty bundle of items. Of course, this is impractical for anything other than very small auctions. Hence, it becomes important to have a good *bidding language* in which bidders can express their valuation functions concisely. Unfortunately, regardless of the language used, there will always be *some* valuation functions that require exponential space to express, for purely information theoretic reasons. However, a good bidding language allows bidders to express *natural* valuation functions concisely. This is analogous to, for example, Bayes Nets: any distribution can be represented using a Bayes Net, but in general, doing so requires specifying an exponentially large number of probabilities. However, natural distributions display a significant amount of conditional independence, and such distributions can be expressed much more concisely using a Bayes Net.

So, the question becomes: which valuation functions are natural? One common assumption is that the bidder is *single-minded*, that is, there is a bundle S_i and a constant k_i such that $v_i(S) = k_i$ if $S_i \subseteq S$, and $v_i(S) = 0$ if $S_i \not\subseteq S$. That is, the bidder has her heart set on a particular bundle; if she gets it, she receives a utility of k_i (and any additional items that she receives will simply be thrown away), but if even one item from S_i is missing, the bundle becomes worthless to her. A single-minded valuation function is easy to express: in this case, a bid consists simply of a bundle S_i and a value k_i . However, usually, bidders are interested in more than one bundle. The *OR-language* allows a bidder to bid on multiple bundles. An example bid in this language is $(\{a\}, 3)$ OR $(\{b, c\}, 4)$ OR $(\{c, d\}, 5)$. This bid indicates that if the bidder receives the bundle $\{a, b, c\}$, her value is 7 (because she receives the first two bundles in her bid); if she receives $\{b, c, d\}$, her utility is 5 (each of the last two bundles in her bid are contained in the bundle she receives, but c can only be counted towards one of them, and the last bundle has the greater valuation). A disadvantage of the OR-language is that it cannot express every valuation function. For example, consider the valuation function $v(\{a\}) = 1, v(\{b\}) = 1, v(\{a, b\}) = 1$ (the bidder wants either item and has no use for a second item). If the bidder bids $(\{a\}, 1)$ OR $(\{b\}, 1)$, this implies a valuation of 2 for the bundle $\{a, b\}$; on the other hand, if she does not include both singleton bundles in her bid, then the bid will not reflect her valuation for those.

An alternative is the *XOR-language*. At most one of the bundles in an XOR-bid can be counted: for example, the bid $(\{a\}, 3)$ XOR $(\{b, c\}, 4)$ XOR $(\{c, d\}, 5)$ indicates that the bidder's value for $\{a, b, c\}$ is 4 (only one of the first two bundles can be counted). In this language, any valuation function can be expressed (if necessary, by XOR-ing all possible bundles together). One downside of the XOR-language is that some natural valuation functions require exponential space to represent—for example, the function $v(S) = |S|$ (which can be concisely represented in the OR language by listing each singleton bundle at a value of 1). Of course,

nothing prevents us from using ORs and XORs simultaneously, to obtain the best of both worlds.

There are also bidding languages with a different flavor. For example, the bidder can specify interactions among the items (Chevalerey *et al.* 2004; Conitzer, Sandholm, & Santi 2005). If a bidder reports a valuation of 2 for item a , a valuation of 3 for item b , and an interaction of -1 between the items, this implies a valuation of $2+3-1 = 4$ for the bundle $\{a, b\}$.

Voting

In a voting setting, the straightforward way for a voter to communicate her preferences is to simply communicate the position of each alternative in her ranking. If there are m alternatives (and hence m different positions), specifying one position requires $O(\log m)$ bits, leading to a total space requirement of $O(m \log m)$ bits. In many settings, this is quite manageable, and because of this the problem of how preferences are represented often does not receive much attention in voting.

Nevertheless, there are many important settings in which there are exponentially many alternatives, so that $m \log m$ space is no longer manageable. For example, in some settings, the set of alternatives may be written as $C = X_1 \times X_2 \times \dots \times X_p$. Here, each X_j corresponds to a separate *issue* on which we need to make a decision. In such a domain, we need to make use of a more sophisticated language for representing preferences, such as a CP-net (Boutilier *et al.* 2004). Preferences expressed in such a language do not always give enough information to recover the full ranking of all alternatives, so not all the standard rules can be applied in such a setting. Some very recent work has been devoted to determining how winners should be chosen in such settings (Lang 2007; Xia, Lang, & Ying 2007a; 2007b), but much more remains to be done here.

Winner determination

Once we have defined the rules by which an outcome is chosen, as well as the language in which preferences are reported, we have a well-defined computational problem of deciding what the outcome is given the reported preferences. In combinatorial auctions, this computational problem is usually called the *winner determination problem*. Since in voting, we are also determining a winner, we will use the same name for the problem in that setting.

Combinatorial auctions

The winner determination problem in a combinatorial auction is to, given the bids (expressed in some bidding language), determine the allocation that maximizes the total value. For simplicity, let us first assume that every bidder i is single-minded, and hence her bid can be represented as (S_i, k_i) . The problem is then to select a subset A of the bids to accept, to maximize $\sum_{i \in A} k_i$, under the constraint that $S_i \cap S_j = \emptyset$ for all $i, j \in A$, $i \neq j$. This problem is NP-hard (Rothkopf, Pekeč, & Harstad 1998), even to approximate (Sandholm 2002). There are various approaches to solving it nonetheless: modeling it as an integer program,

writing a search-based algorithm (Sandholm 2006), or using a dynamic programming algorithm (Rothkopf, Pekeč, & Harstad 1998).

It turns out that any algorithm for the winner determination problem with single-minded bids can be extended to deal with bids that use ORs and XORs. For ORs, this is easy to see: as far as winner determination is concerned, there is no difference between receiving two single-minded bids (S_1, k_1) and (S_2, k_2) , or receiving a single bid (S_1, k_1) OR (S_2, k_2) , because in either case the auctioneer can award both bundles for a value of $k_1 + k_2$ if and only if the bundles do not overlap. This argument does not work for XORs, but there is a clever trick for converting an XOR into an OR (Fujishima, Leyton-Brown, & Shoham 1999; Nisan 2000): given a bid (S_1, k_1) XOR (S_2, k_2) , create a dummy item d , and replace the bid by $(S_1 \cup \{d\}, k_1)$ OR $(S_2 \cup \{d\}, k_2)$. Now the two bundles can no longer both be awarded to the bidder, because they overlap. Because of this, most research on the combinatorial auction winner determination problem has focused on single-minded bids.

While the winner determination problem with single-minded bids is NP-hard in general, it can become polynomial-time solvable if the bids lie in a restricted class (Rothkopf, Pekeč, & Harstad 1998; Tennenholtz 2000; Penn & Tennenholtz 2000; Sandholm & Suri 2003; Conitzer, Derryberry, & Sandholm 2004; Gottlob & Greco 2007). For example, it is polynomial-time solvable if each bid is on at most two items (Rothkopf, Pekeč, & Harstad 1998), or if each bid is on a connected set of vertices in a graph with bounded treewidth (Conitzer, Derryberry, & Sandholm 2004) (this result can be generalized further (Gottlob & Greco 2007)).

For other bidding languages, the complexity of the winner determination problem may be different. For example, for the language based on specifying interactions among items (described above), the winner determination problem is NP-hard even with only pairwise interactions.

Voting

In settings where m is not extremely large, so that each voter can communicate her complete ranking of all the alternatives, determining the winning alternative is computationally straightforward for most voting rules. For example, a rule such as Borda requires nothing more than adding up the scores of the alternatives. However, this is not the case for all voting rules: some of them are in fact NP-hard to execute (Bartholdi, Tovey, & Trick 1989b; Hemaspaandra, Hemaspaandra, & Rothe 1997; Cohen, Schapire, & Singer 1999; Dwork *et al.* 2001; Rothe, Spakowski, & Vogel 2003; Ailon, Charikar, & Newman 2005; Alon 2006; Conitzer 2006; Procaccia, Rosenschein, & Zohar 2007; Brandt, Fischer, & Harrenstein 2007). As an example, let us take the Slater rule, which requires finding a ranking that is inconsistent with the outcomes of as few pairwise elections as possible. Determining the outcomes of the pairwise elections is easy. It is helpful to summarize this information in a directed graph, in which the alternatives are the vertices and each edge points from the winner of a pairwise election to the loser. If this graph is acyclic,

then it already corresponds to a ranking. If not, then the Slater rule asks us to make the graph acyclic, by reversing as few edges as possible. This problem turns out to be NP-hard (Ailon, Charikar, & Newman 2005; Alon 2006; Conitzer 2006). The Kemeny rule can similarly be interpreted in a graph-theoretic way: to do so, we use the same graph as for the Slater rule, except we add a weight to each edge (a, b) which is equal to the margin of a 's pairwise victory over b . The problem now becomes to reverse a set of edges of minimum total *weight* to make the graph acyclic. It is possible to find votes so that all the weights become equal to each other, so the Kemeny problem is at least as hard as the Slater problem, that is, NP-hard. Still, it is possible to solve reasonably-sized winner determination instances for the Kemeny rule, using search-based techniques (Davenport & Kalagnanam 2004) or integer programming (Conitzer, Davenport, & Kalagnanam 2006).

Preference elicitation

So far, we have assumed that each agent reports her complete preferences all at once. While a good language for describing preferences can be very helpful in doing so, the agent still needs to determine her entire preferences before she can report them. An alternative approach is to use a sequential process, where one agent reports some information about her preferences; then, based on that information, another agent reports some information about her preferences, *etc.*, until we know enough to determine the outcome. This approach is known as *preference elicitation*, and it is usually guided by a central party, the *elicitor*, who determines who has to report which information next (usually by asking that agent a *query* about her preferences).

When we use preference elicitation, it is generally not necessary for each agent to reveal all of her preferences. Very often, once we have obtained some preference information from the agents, other information becomes irrelevant. For example, if we have already determined that one agent's value for a particular bundle is not high enough to have any chance of winning, then there may not be any need to determine the exact value. As another example, if we have already determined that one alternative has no chance of winning, then there may not be any need to determine its exact position in a voter's preferences. Making use of such observations can reduce the amount of information that the agents have to communicate. However, the benefit here is more than mere communicational convenience. Often, the greater benefit is that the agent does not even need to *determine* her preferences completely. A significant amount of deliberation effort can be required to determine even a single bit of preference information (for example, which of two alternatives is preferred), so it can be very valuable to realize that this information is not needed. An additional benefit is that the agents' privacy is improved, in the sense that they have to release less of their preference information. It follows that preference elicitation can be extremely valuable even in settings where preferences can be described concisely—for example, voting over a small set of alternatives.

Combinatorial auctions

In combinatorial auctions, it is easy to see the potential benefits of preference elicitation. Even when armed with the best bidding language, determining one's exact valuation for every bundle can still be overwhelming. It is therefore not surprising that much of the work on preference elicitation has focused on combinatorial auctions.

In a sense, single-item auctions such as the English, Japanese, and Dutch auctions are already doing some preference elicitation. At each stage, bidders reveal some information, such as whether the current price is too high for them; and once we know the winning bidder, we end the auction, even if we do not yet know all bidders' exact valuations. Indeed, some of the approaches to preference elicitation in combinatorial auctions try to mimic these single-item auctions. Several *ascending* combinatorial auctions have been proposed (Parkes & Ungar 2000; Ausubel & Milgrom 2002; Parkes 2006). In every round of an ascending combinatorial auction, each bidder faces a price for each bundle, and needs to decide which bundle she would like to buy under these prices. The prices start low, so that bidders will choose large bundles that overlap with each other. Each round, the prices are raised, up to the point that the chosen bundles no longer overlap with each other. While ascending combinatorial auctions can, under certain conditions, result in the optimal allocation, they also face some inherent limitations (Blumrosen & Nisan 2005).

Ascending combinatorial auctions typically proceed in a very systematic fashion. It is also possible to take a more flexible preference elicitation approach, where the auctioneer/elicitor can ask any agent any query at any point. Common queries include the *demand query*, where a bidder is asked which bundle she would prefer under certain prices (as in the ascending auctions), and the *value query*, which simply asks for the bidder's valuation for a specific bundle.

One question that is often studied in preference elicitation is the following: if a bidder's valuation function is guaranteed to lie in a certain class of functions C , can we determine the bidder's valuation function exactly using only a polynomial number of queries? A number of positive results have been obtained along this line. For example, a bidder's valuation function can be elicited exactly with a number of value and demand queries that is polynomial in the length of the valuation function's representation in the XOR-language (Lahaie & Parkes 2004). Hence, valuations that admit a concise XOR-representation can be elicited efficiently using such queries. However, this result only holds if the demand queries can set prices on bundles (rather than just on individual items) (Blum *et al.* 2004). There are numerous other results that show that various classes can(not) be elicited using a polynomial number of queries of a given type (Zinkevich, Blum, & Sandholm 2003; Santi, Conitzer, & Sandholm 2004; Conitzer, Sandholm, & Santi 2005; Blumrosen & Nisan 2005; Lahaie, Constantin, & Parkes 2005).

If there are no restrictions on the valuation function, some negative results are known. For example, it has been shown using tools from communication complexity theory (Kushilevitz & Nisan 1997) that the winner determi-

nation problem in general requires exponential communication (Nisan & Segal 2005). This is true regardless of the types of query that are allowed.

Voting

When agents are voting over a set of alternatives that is not extremely large, the potential benefits of preference elicitation are less dramatic than in a combinatorial auction: even reporting one's complete preferences requires only a polynomial amount of communication. Nevertheless, as we have already noted, determining these preferences generally still requires a large amount of deliberation effort. Hence, there is still significant value in doing preference elicitation in such settings.

As in the case of combinatorial auctions, there has been some work on determining how many queries are needed to completely elicit a voter's preferences, assuming that the preferences lie in a specific class. For instance: sometimes there is a natural order $<$ on the alternatives—for example, some candidates may be more “right-wing” than others. A voter's preferences \succ_i are said to be *single-peaked* with respect to this order if, whenever $a < b < c$ or $c < b < a$ (where a is i 's most-preferred alternative), we have that $b \succ_i c$. That is, the voter always prefers alternatives that are closer to her most-preferred alternative. It has been shown that single-peaked preferences can be elicited using a linear number of *comparison queries* (which ask the voter which of two alternatives is preferred), if either the order $<$ is known, or at least one other vote that is single-peaked with respect to $<$ is known (Conitzer 2007b).

Another important topic is the communication complexity of executing various rules—that is, for a given rule, what is the minimum number of bits that needs to be communicated by the voters (in the worst case) to determine the winner? It should be noted that here, there is no constraint on the types of query that are used. Some of the rules, such as Borda and Copeland, turn out to require $\Omega(nm \log m)$ communication in the worst case—that is, in the worst case, the communication requirements are roughly as bad as having everyone reveal all of their preferences. For other rules, however, we can get away with much less communication: for example, STV requires only $O(n(\log m)^2)$ (using the straightforward protocol where everyone reports their most-preferred alternative first; then, when an alternative is eliminated, all voters who ranked that alternative first communicate their next-most preferred alternative, *etc.*), and Bucklin requires only $O(nm)$ (using a more sophisticated protocol based on binary search) (Conitzer & Sandholm 2005b).

Another key computational problem for preference elicitation is the following (Conitzer & Sandholm 2002; Konczak & Lang 2005; Pini *et al.* 2007; Walsh 2007): given some of the preferences of some of the voters, do we already know which alternative is the winner? If we can solve this problem, then we will know when we can stop eliciting preferences and declare the winner. This problem is often easy, but sometimes it is NP-hard, for example for STV (Conitzer & Sandholm 2002).

A final direction is to optimize the elicitation process: given a prior distribution over the voters' preferences, plan

the elicitation process to minimize the expected number of voters whose preferences we elicit before we know the winner. For most rules, this problem is NP-hard even if we are completely certain beforehand about how voters will vote (but we still need to elicit their preferences to prove that we are right) (Conitzer & Sandholm 2002).

Strategic agents and mechanism design

A final important issue is that of strategic bidding/voting. Depending on the rules for the auction/election and the bids/votes cast by the other agents, it may not be in an agent's best interest to truthfully report her preferences. An agent that reports preferences other than the preferences she truly has (with the aim of obtaining a better outcome for herself) is said to *misreport* or *manipulate*. If agents manipulate, this can be detrimental to the quality of the outcome: even if we manage to solve the winner determination problem to optimality with respect to the reported preferences, there is no guarantee that this results in an outcome that is good with respect to the *true* preferences, which is what we really care about. As we will see shortly, the approaches that are taken to address this problem are generally different in combinatorial auctions than in voting. This is because in combinatorial auctions, the payments can be used to remove any incentive for the bidders to misreport their valuation function, whereas in general voting settings, incentives to misreport always remain.

Combinatorial auctions

To illustrate the main idea, it is helpful to return to the single-item auction setting. We assume that a bidder i 's utility for winning the item at price π_i is $v_i - \pi_i$, where v_i is the bidder's (true) valuation for the item; and the utility for not winning is 0. Let us first consider the first-price sealed-bid auction. It is easy to see that in this auction, it never makes sense for a bidder to bid her true valuation v_i , because in this case, even if she wins, her utility is $v_i - v_i = 0$. Rather, a bidder needs to bid lower than her true valuation to have any chance of obtaining positive utility.

On the other hand, let us consider the second-price sealed-bid auction. To see how a bidder i should bid in this auction, let us first suppose that she knows all the other bids. Then, she has only two choices: either bid higher than the highest other bid, b^* , and obtain utility $v_i - b^*$; or bid lower, and obtain utility 0. Clearly, she should do the former if and only if $v_i > b^*$. But this can be achieved simply by bidding truthfully (for which she does not even need to know the others' bids). Thus, it is always optimal for a bidder to reveal her true valuation in a second-price sealed-bid auction. That is, this auction is (*dominant-strategies*) *incentive compatible*.

A natural question is whether it is possible to make combinatorial auctions incentive compatible as well by using the right payments. It turns out that this is indeed possible, by using the *Generalized Vickrey Auction (GVA)* (sometimes also referred to as the *Clarke mechanism*, or the *VCG mechanism* (Vickrey 1961; Clarke 1971; Groves 1973)). The GVA works as follows: first, solve the winner determination problem, to obtain an optimal allocation S_1, \dots, S_n , with

total value $V = \sum_{i=1}^n v_i(S_i)$. Now, to determine how much bidder i must pay, remove i from the auction, and solve the winner determination problem again with the remaining bidders, obtaining a total utility of V_{-i} (which is at most V). Then, bidder i must pay $V_{-i} - (V - v_i(S_i))$. It is not difficult to show that the GVA is incentive compatible.

While the GVA appears to solve the problem of strategic bidding nicely, it still has a number of drawbacks. First of all, if obtaining the efficient allocation of resources is not the objective, then the GVA may not be optimal. (For example, there has been significant interest in determining the combinatorial auction mechanism that maximizes expected revenue, although this turns out to be surprisingly difficult (Avery & Hendershott 2000; Armstrong 2000; Conitzer & Sandholm 2004; Likhodedov & Sandholm 2004; 2005).) Another problem is that the GVA is very vulnerable to multiple bidders *colluding*: whereas there is no incentive for a single bidder to misreport by herself, a group of bidders can make themselves (sometimes much) better off by collectively misreporting (Ausubel & Milgrom 2006; Conitzer & Sandholm 2006a). A bidder can also make herself much better off by pretending to be multiple distinct bidders, which is often possible in open, anonymous environments such as the Internet. That is, the GVA is not *false-name-proof*. Several combinatorial auction mechanisms have been proposed that are false-name-proof, but this comes at the cost of efficiency (Yokoo, Sakurai, & Matsubara 2001; Yokoo 2003; Yokoo, Sakurai, & Matsubara 2004; Yokoo, Matsutani, & Iwasaki 2006). (An alternative approach is to verify the identities of some of the bidders after the fact (Conitzer 2007c).) A final issue is that the winner determination problems must be solved to optimality: if an approximately optimal solution is used, the GVA is in general no longer incentive compatible. A significant body of research has focused on creating approximation algorithms for the winner determination problem that can be combined with payment schemes that make them incentive compatible (Nisan & Ronen 2001; Lehmann, O'Callaghan, & Shoham 2002; Mu'alem & Nisan 2002; Bartal, Gonen, & Nisan 2003; Archer *et al.* 2003; Dobzinski, Nisan, & Schapira 2006; Dobzinski & Nisan 2007a; 2007b).

Voting

While the GVA mechanism solves the problem of strategic bidding in combinatorial auctions (with some serious caveats), in voting settings no such solution exists. (This is fundamentally due to the lack of payments in voting settings: the Clarke mechanism, of which the GVA is a special case, can also be applied to voting settings in which the voters can be required to make payments.) Specifically, there is the *Gibbard-Satterthwaite impossibility theorem* (Gibbard 1973; Satterthwaite 1975), which states that with 3 or more alternatives, every voting rule that is incentive compatible is either dictatorial, or is such that there is an alternative that cannot win, regardless of the votes. (One may wonder if one can do better using voting rules that use randomization in their choice of the winner; however, a later paper by Gibbard (Gibbard 1977) characterizes the class of incentive compatible randomized voting rules completely,

and the result is still mostly negative in that all the rules in the class have some undesirable properties. As an aside, the class becomes even smaller if we require false-name-proofness (Conitzer 2007a).)

Still, something must be done to address the problem of strategic voting. Several recent papers have pursued the approach of using computational hardness as a barrier to manipulation. The idea here is the following: while for any reasonable rule, there are guaranteed to be instances where there exists a beneficial manipulation (that is, a way of voting insincerely that makes the voter better off), the voter will not be able to make use of this if the manipulation is computationally too hard to discover. Indeed, there has been a number of results showing that the problem of finding a beneficial manipulation is NP-hard (for some rules, for some definitions of the manipulation problem) (Bartholdi, Tovey, & Trick 1989a; Bartholdi & Orlin 1991; Conitzer & Sandholm 2003; Elkind & Lipmaa 2005; Hemaspaandra & Hemaspaandra 2007; Conitzer, Sandholm, & Lang 2007; Procaccia, Rosenschein, & Zohar 2007). While this shows that there is indeed some computational barrier to manipulation, the result is not as strong as we would like. Specifically, NP-hardness is a worst-case measure of hardness: it shows that (assuming $P \neq NP$) any algorithm for finding manipulations is going to require exponential time on *some* instances, but it could still be the case that the algorithm rapidly identifies a beneficial manipulation on most instances. It would be more satisfactory to have a result that shows that *most* instances are hard to manipulate. Unfortunately, recently some negative results have been obtained that show that, under some restrictions on the rule, it is usually easy to find a manipulation (Conitzer & Sandholm 2006b; Procaccia & Rosenschein 2007a; 2007b). Still, it seems that the book is not yet closed on using computational hardness as a barrier to manipulation. Incidentally, computational hardness has also been considered as a barrier against other types of undesirable behavior, for example by the chairperson of the election (Bartholdi, Tovey, & Trick 1992; Hemaspaandra, Hemaspaandra, & Rothe 2005; Faliszewski, Hemaspaandra, & Hemaspaandra 2006; Procaccia, Rosenschein, & Zohar 2007).

Another issue is that if we use preference elicitation on a voting rule that is not incentive compatible, this may introduce additional opportunities for manipulation. This is because from the queries that a voter is asked, she can infer something about how the others are voting; and she may wish to change her vote based on this information (Conitzer & Sandholm 2002). This is an issue that does not occur if we use a (dominant-strategies) incentive compatible mechanism (such as the GVA): in that case, answering the queries truthfully becomes an *ex post* equilibrium. That is, it is optimal for every agent to answer the queries truthfully, as long as the others do so as well—regardless of what their preferences are.

Conclusions

In this paper, we considered several issues that are of importance both in combinatorial auctions and in voting settings,

and compared the research on these issues in the two domains.

We first considered the rules for how outcomes are chosen in combinatorial auctions and in voting. Here we noticed a major difference. Most (though not all) combinatorial auction designs agree on the objective of maximizing the efficiency of the allocation of items, and the differences among the designs are primarily due to other factors, such as how bidders' valuations are elicited and what incentives bidders have when they bid strategically. In contrast, in voting settings, it is generally not clear what the ideal objective is, and different voting rules correspond to different objectives.

We then considered the languages in which agents express their preferences. In combinatorial auctions, there is a significant body of research on various bidding languages and their strengths and weaknesses. In voting, there is not much research on languages yet; this is presumably primarily because most work so far has dealt with settings where the number of alternatives is not extremely large, and in such settings it does not require much space to express any ranking. However, this is not true in the case of combinatorial alternative spaces (which can have exponential size). Some languages for representing preferences have been proposed for combinatorial alternative spaces, and as the interest in such settings grows, undoubtedly so will the interest in voting languages.

Subsequently, we considered the winner determination problem. In combinatorial auctions, the winner determination problem has received a lot of attention, in part because this is perhaps the first problem that needs to be solved to be able to run a combinatorial auction. In voting, when the number of alternatives is not extremely large, the winner determination problem is quite easy for most rules. But there are some rules for which it is NP-hard, and various techniques have been proposed for running these rules nonetheless. It seems that as the focus shifts to combinatorial alternative spaces, more difficult winner determination problems will emerge.

We then considered preference elicitation, that is, sequentially querying the agents for parts of their preferences until we know enough to determine the outcome. Again, this is a problem that has received more attention in combinatorial auctions, presumably because in general, a bidder must communicate exponentially many values, so any reduction in this communication is highly desirable. Nevertheless, preference elicitation has also received some attention in voting settings. This is because even though specifying a ranking requires space polynomial in the number of alternatives, it generally requires significant deliberation effort to decide on the precise ranking, and preference elicitation can reduce the required deliberation effort. Yet again, it seems that preference elicitation will receive more attention in settings with exponentially many alternatives.

Finally, we considered strategic agents, who will misreport their preferences if this is to their benefit. Here there is a major difference between combinatorial auctions and voting. In combinatorial auctions, it is possible to remove any incentive to misreport, using techniques from mechanism design. Unfortunately, in voting settings, mechanism

design provides little more than negative results, and hence research has turned to other ways to prevent misreporting—specifically, making it computationally hard to do so beneficially. However, both in combinatorial auctions and voting, there is a growing interest in addressing new types of manipulation—specifically, the ability to participate in the mechanism multiple times under different identifiers. It appears that this type of manipulation is more difficult to address in combinatorial auctions, so perhaps the techniques that will be designed to address this will be similar across the two domains.

In summary, many of the same issues are at play in combinatorial auctions and voting settings. Considering current trends in research in the two domains, it seems that the similarities will only grow. Hence, it appears that researchers in each domain would benefit from keeping a close eye on the other domain.

References

- Ailon, N.; Charikar, M.; and Newman, A. 2005. Aggregating inconsistent information: Ranking and clustering. *STOC*.
- Alon, N. 2006. Ranking tournaments. *SIAM Journal of Discrete Mathematics* 20:137–142.
- Archer, A.; Papadimitriou, C. H.; Talwar, K.; and Tardos, E. 2003. An approximate truthful mechanism for combinatorial auctions with single parameter agents. *SODA*.
- Armstrong, M. 2000. Optimal multi-object auctions. *Review of Economic Studies* 67:455–481.
- Arrow, K. 1963. *Social choice and individual values*. New Haven: Cowles Foundation, 2nd edition. 1st edition 1951.
- Ausubel, L., and Milgrom, P. 2002. Ascending auctions with package bidding. *Frontiers of Theoretical Economics* 1. No. 1, Article 1.
- Ausubel, L. M., and Milgrom, P. 2006. The lovely but lonely Vickrey auction. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press, chapter 1.
- Avery, C., and Hendershott, T. 2000. Bundling and optimal auctions of multiple products. *Review of Economic Studies* 67:483–497.
- Bailey, M. J. 1997. The demand revealing process: to distribute the surplus. *Public Choice* 91:107–126.
- Bartal, Y.; Gonen, R.; and Nisan, N. 2003. Incentive compatible multi-unit combinatorial auctions. *TARK*.
- Bartholdi, III, J., and Orlin, J. 1991. Single transferable vote resists strategic voting. *Social Choice and Welfare* 8(4):341–354.
- Bartholdi, III, J.; Tovey, C.; and Trick, M. 1989a. The computational difficulty of manipulating an election. *Social Choice and Welfare* 6(3):227–241.
- Bartholdi, III, 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* 6:157–165.
- Bartholdi, III, J.; Tovey, C.; and Trick, M. 1992. How hard is it to control an election? *Math. Comput. Modelling* 16(8-9):27–40. Formal theories of politics, II.
- Blum, A.; Jackson, J.; Sandholm, T.; and Zinkevich, M.

2004. Preference elicitation and query learning. *JMLR* 5:649–667.
- Blumrosen, L., and Nisan, N. 2005. On the computational power of iterative auctions. *EC*, 29–43.
- Boutilier, C.; Brafman, R.; Domshlak, C.; Hoos, H.; and Poole, D. 2004. CP-nets: a tool for representing and reasoning with conditional ceteris paribus statements. *JAIR* 21:135–191.
- Brandt, F.; Fischer, F.; and Harrenstein, P. 2007. The computational complexity of choice sets. *TARK*.
- Cavallo, R. 2006. Optimal decision-making with minimal waste: Strategyproof redistribution of VCG payments. *AAMAS*, 882–889.
- Chevaleyre, Y.; Endriss, U.; Estivie, S.; and Maudet, N. 2004. Multiagent resource allocation with k-additive utility functions. In *Workshop on Computer Science and Decision Theory*.
- Clarke, E. H. 1971. Multipart pricing of public goods. *Public Choice* 11:17–33.
- Cohen, W.; Schapire, R.; and Singer, Y. 1999. Learning to order things. *JAIR* 10:213–270.
- Conitzer, V., and Sandholm, T. 2002. Vote elicitation: Complexity and strategy-proofness. *AAAI*, 392–397.
- Conitzer, V., and Sandholm, T. 2003. Universal voting protocol tweaks to make manipulation hard. *IJCAI*, 781–788.
- Conitzer, V., and Sandholm, T. 2004. Self-interested automated mechanism design and implications for optimal combinatorial auctions. *EC*, 132–141.
- Conitzer, V., and Sandholm, T. 2005a. Common voting rules as maximum likelihood estimators. *UAI*, 145–152.
- Conitzer, V., and Sandholm, T. 2005b. Communication complexity of common voting rules. *EC*, 78–87.
- Conitzer, V., and Sandholm, T. 2006a. Failures of the VCG mechanism in combinatorial auctions and exchanges. *AAMAS*, 521–528.
- Conitzer, V., and Sandholm, T. 2006b. Nonexistence of voting rules that are usually hard to manipulate. *AAAI*.
- Conitzer, V.; Davenport, A.; and Kalagnanam, J. 2006. Improved bounds for computing Kemeny rankings. *AAAI*.
- Conitzer, V.; Derryberry, J.; and Sandholm, T. 2004. Combinatorial auctions with structured item graphs. *AAAI*, 212–218.
- Conitzer, V.; Sandholm, T.; and Lang, J. 2007. When are elections with few candidates hard to manipulate? *Journal of the ACM* 54(3):Article 14, 1–33.
- Conitzer, V.; Sandholm, T.; and Santi, P. 2005. Combinatorial auctions with k -wise dependent valuations. *AAAI*, 248–254.
- Conitzer, V. 2006. Computing Slater rankings using similarities among candidates. *AAAI*.
- Conitzer, V. 2007a. Anonymity-proof voting rules. Draft.
- Conitzer, V. 2007b. Eliciting single-peaked preferences using comparison queries. *AAMAS*, 408–415.
- Conitzer, V. 2007c. Limited verification of identities to induce false-name-proofness. *TARK*, 102–111.
- Davenport, A., and Kalagnanam, J. 2004. A computational study of the Kemeny rule for preference aggregation. *AAAI*, 697–702.
- de Caritat (Marquis de Condorcet), M. J. A. N. 1785. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris: L'Imprimerie Royale.
- Dobzinski, S., and Nisan, N. 2007a. Limitations of vcg-based mechanisms. *STOC*, 338–344.
- Dobzinski, S., and Nisan, N. 2007b. Mechanisms for multi-unit auctions. *EC*, 346–351.
- Dobzinski, S.; Nisan, N.; and Schapira, M. 2006. Truthful randomized mechanisms for combinatorial auctions. *STOC*, 644–652.
- Dwork, C.; Kumar, R.; Naor, M.; and Sivakumar, D. 2001. Rank aggregation methods for the web. *WWW*, 613–622.
- Elkind, E., and Lipmaa, H. 2005. Hybrid voting protocols and hardness of manipulation. *ISAAC*.
- Faliszewski, P.; Hemaspaandra, E.; and Hemaspaandra, L. A. 2006. The complexity of bribery in elections. *AAAI*.
- Fujishima, Y.; Leyton-Brown, K.; and Shoham, Y. 1999. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. *IJCAI*, 548–553.
- Gibbard, A. 1973. Manipulation of voting schemes: a general result. *Econometrica* 41:587–602.
- Gibbard, A. 1977. Manipulation of schemes that mix voting with chance. *Econometrica* 45:665–681.
- Gottlob, G., and Greco, G. 2007. On the complexity of combinatorial auctions: Structured item graphs and hypertree decompositions. *EC*, 152–161.
- Groves, T. 1973. Incentives in teams. *Econometrica* 41:617–631.
- Guo, M., and Conitzer, V. 2007. Worst-case optimal redistribution of VCG payments. *EC*, 30–39.
- Hemaspaandra, E., and Hemaspaandra, L. A. 2007. Dichotomy for voting systems. *JCSS* 73(1):73–83.
- 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.
- Hemaspaandra, E.; Hemaspaandra, L. A.; and Rothe, J. 2005. Anyone but him: The complexity of precluding an alternative. *AAAI*.
- Konczak, K., and Lang, J. 2005. Voting procedures with incomplete preferences. In *Multidisciplinary Workshop on Advances in Preference Handling*.
- Kushilevitz, E., and Nisan, N. 1997. *Communication Complexity*. Cambridge University Press.
- Lahaie, S., and Parkes, D. 2004. Applying learning algorithms to preference elicitation. *EC*, 180–188.
- Lahaie, S.; Constantin, F.; and Parkes, D. C. 2005. More on the power of demand queries in combinatorial auctions: Learning atomic languages and handling incentives. *IJCAI*, 959–964.
- Lang, J. 2007. Vote and aggregation in combinatorial domains with structured preferences. *IJCAI*, 1366–1371.
- Lehmann, D.; O'Callaghan, L. I.; and Shoham, Y. 2002. Truth revelation in rapid, approximately efficient combinatorial auctions. *Journal of the ACM* 49(5):577–602.
- Likhodedov, A., and Sandholm, T. 2004. Methods for boosting revenue in combinatorial auctions. *AAAI*, 232–237.

- Likhodedov, A., and Sandholm, T. 2005. Approximating revenue-maximizing combinatorial auctions. *AAAI*.
- Moulin, H. 2007. Efficient, strategy-proof and almost budget-balanced assignment. Working Paper.
- Mu'alem, A., and Nisan, N. 2002. Truthful approximate mechanisms for restricted combinatorial auctions. *AAAI*, 379–384.
- Myerson, R. 1981. Optimal auction design. *Mathematics of Operations Research* 6:58–73.
- Nisan, N., and Ronen, A. 2001. Algorithmic mechanism design. *Games and Economic Behavior* 35:166–196.
- Nisan, N., and Segal, I. 2005. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*.
- Nisan, N. 2000. Bidding and allocation in combinatorial auctions. *EC*, 1–12.
- Parkes, D., and Ungar, L. 2000. Iterative combinatorial auctions: Theory and practice. *AAAI*, 74–81.
- Parkes, D. 2006. Iterative combinatorial auctions. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press. chapter 3.
- Penn, M., and Tennenholtz, M. 2000. Constrained multi-object auctions and b -matching. *Information Processing Letters* 75(1–2):29–34.
- Pini, M. S.; Rossi, F.; Venable, K. B.; and Walsh, T. 2005. Aggregating partially ordered preferences: possibility and impossibility results. *TARK*.
- Pini, M. S.; Rossi, F.; Venable, K. B.; and Walsh, T. 2007. Incompleteness and incomparability in preference aggregation. *IJCAI*.
- Porter, R.; Shoham, Y.; and Tennenholtz, M. 2004. Fair imposition. *Journal of Economic Theory* 118:209–228.
- Procaccia, A. D., and Rosenschein, J. S. 2007a. Average-case tractability of manipulation in voting via the fraction of manipulators. *AAMAS*.
- Procaccia, A. D., and Rosenschein, J. S. 2007b. Junta distributions and the average-case complexity of manipulating elections. *JAIR* 28:157–181.
- Procaccia, A. D.; Rosenschein, J. S.; and Zohar, A. 2007. Multi-winner elections: Complexity of manipulation, control and winner-determination. *IJCAI*.
- Rossi, F.; Pini, M. S.; Venable, K. B.; and Walsh, T. 2006. Strategic voting when aggregating partially ordered preferences. *AAMAS*, 685–687.
- Rothe, J.; Spakowski, H.; and Vogel, J. 2003. Exact complexity of the winner problem for Young elections. In *Theory of Computing Systems*, volume 36(4). Springer-Verlag. 375–386.
- Rothkopf, M.; Pekeč, A.; and Harstad, R. 1998. Computationally manageable combinatorial auctions. *Management Science* 44(8):1131–1147.
- Sandholm, T., and Suri, S. 2003. BOB: Improved winner determination in combinatorial auctions and generalizations. *AIJ* 145:33–58.
- Sandholm, T.; Suri, S.; Gilpin, A.; and Levine, D. 2002. Winner determination in combinatorial auction generalizations. *AAMAS*, 69–76.
- Sandholm, T. 2002. Algorithm for optimal winner determination in combinatorial auctions. *AIJ* 135:1–54.
- Sandholm, T. 2006. Optimal winner determination algorithms. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press. 337–368.
- Santi, P.; Conitzer, V.; and Sandholm, T. 2004. Towards a characterization of polynomial preference elicitation with value queries in combinatorial auctions. *COLT*, 1–16.
- Satterthwaite, M. 1975. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory* 10:187–217.
- Tennenholtz, M. 2000. Some tractable combinatorial auctions. *AAAI*.
- Vickrey, W. 1961. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance* 16:8–37.
- Walsh, T. 2007. Uncertainty in preference elicitation and aggregation. *AAAI*.
- Xia, L.; Lang, J.; and Ying, M. 2007a. Sequential voting rules and multiple elections paradoxes. *TARK*.
- Xia, L.; Lang, J.; and Ying, M. 2007b. Strongly decomposable voting rules on multiattribute domains. *AAAI*.
- Yokoo, M.; Matsutani, T.; and Iwasaki, A. 2006. False-name-proof combinatorial auction protocol: Groves mechanism with submodular approximation. *AAMAS*, 1135–1142.
- Yokoo, M.; Sakurai, Y.; and Matsubara, S. 2001. Robust combinatorial auction protocol against false-name bids. *AIJ* 130(2):167–181.
- Yokoo, M.; Sakurai, Y.; and Matsubara, S. 2004. The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. *Games and Economic Behavior* 46(1):174–188.
- Yokoo, M. 2003. The characterization of strategy/false-name proof combinatorial auction protocols: Price-oriented, rationing-free protocol. *IJCAI*, 733–742.
- Young, H. P., and Levenglick, A. 1978. A consistent extension of Condorcet's election principle. *SIAM Journal of Applied Mathematics* 35(2):285–300.
- Young, H. P. 1995. Optimal voting rules. *Journal of Economic Perspectives* 9(1):51–64.
- Zinkevich, M.; Blum, A.; and Sandholm, T. 2003. On polynomial-time preference elicitation with value queries. *EC*, 176–185.