Making Markets and Democracy Work: A Story of Incentives and Computing

IJCAI-03 Computers and Thought Award Talk Abstract[*]

Tuomas Sandholm
Carnegie Mellon University
Computer Science Department
5000 Forbes Avenue
Pittsburgh, PA 15213
sandholm@cs.cmu.edu

Abstract

Collective choice settings are the heart of society. Game theory provides a basis for engineering the incentives into the interaction *mechanism* (*e.g.*, rules of an election or auction) so that a desirable system-wide *outcome* (*e.g.*, president, resource allocation, or task allocation) is chosen even though every agent acts based on self-interest.

However, there are a host of computer science issues not traditionally addressed in game theory that have to be addressed in order to make mechanisms work in the real world. Those computing, communication, and privacy issues are deeply intertwined with the economic incentive issues. For example, the fact that agents have limited computational capabilities to determine their own (and others') preferences ruins the incentive properties of established auction mechanisms, and gives rise to new issues. On the positive side, computational complexity can be used as a barrier to strategic behavior in settings where economic mechanism design falls short.

Novel computational approaches also enable new economic institutions. For example, market clearing technology with specialized search algorithms is enabling a form of interaction that I call *expressive competition*. As another example, selective incremental preference elicitation can determine the optimal outcome while requiring the agents to determine and reveal only a small portion of their preferences. Furthermore, *automated mechanism design* can yield better mechanisms than the best known to date.

[*]This material is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IIS-9800994, ITR IIS-0081246, and ITR IIS-0121678.

1 Introduction

Collective choice settings are the heart of society. Citizens voting determines a president, producers and consumers bidding determines a set of trades, and surfers hitting links in web browsers determines a bandwidth allocation. A key difficulty in collective choice is that the agents generally have conflicting preferences over the *outcomes* (*e.g.*, presidents, resource allocations, or task allocations). Work in mechanism design, a subfield of game theory, provides a basis for engineering the incentives into the interaction *mechanism* (*e.g.*, rules of an election or auction) so that a desirable—according to some objective—outcome is chosen even though every party acts based on self-interest.

However, there are a host of computer science issues not traditionally addressed in game theory that have to be addressed in order to make mechanisms work in the real world. Those computing, communication, and privacy issues have to be handled while simultaneously handling the economic incentive issues. This is a particularly exciting research area because those issues are intimately intertwined, as I hope to convey. For example, the fact that agents have limited computational capabilities can ruin the incentive properties of established auction mechanisms, and give rise to new game-theoretic issues. On the positive side, computational complexity can be used as a barrier to insincere strategic behavior in settings where economic incentive engineering is known to fall short.

Novel computational approaches and algorithms can also enable new economic institutions. For example, sophisticated market clearing technology with specialized search algorithms enables a new form of interaction that I call *expressive competition*: empowering market participants with potent expressiveness akin to human-to-human negotiation while at the same time harnessing the forces of competition, the global scale of the Internet, and the speed and accuracy of algorithmic market clearing with all relevant information in hand. Furthermore, even the mechanism itself (such as the rules of an auction) can be designed automatically—in many cases yielding better mechanisms than the best known to date.

This writeup begins from distributed peer-to-peer negotiation (Section 2), and transitions to markets that have have a mediator such as an auction server (Section 3). In this context I will lay out the vision and technology for expressive competition. Section 4 discusses multiagent preference elicitation in auctions and voting settings: can the mediator elicit the information needed to determine the optimal outcome without requiring the agents to determine and reveal their preferences about impertinent aspects of the problem? The issue of how carefully computationally constrained agents should determine their preferences is addressed in Section 5. It turns out that the computational constraint undermines desirable incentive properties in established auction mechanisms, and gives rise to new game-theoretic issues, in particular a phenomenon which I call *strategic computing*: using one's limited computing to approximate others' preferences at the cost of approximating one's own. The reverse is shown in Section 6: computational complexity can be used as a barrier to undesirable strategic behavior. I illustrate this in voting.

Section 7 discusses a new idea which I call *automated mechanism design*: designing the interaction mechanism computationally for the specific setting at hand. Section 8 shows how mechanism design can not only be used to lead to a desirable outcome in a multiagent system, but also to determine a way to *execute* the outcome. In particular, the writeup looks at safe exchange mechanisms for carrying out trades among anonymous parties on the Internet. Finally, conclusions, perspective, and promising avenues for future research are presented.

2 A first tack: Peer-to-peer negotiation

Work on automated negotiation began in peer-to-peer contexts where the negotiating agents (humans or software) make deals with each other. A key insight for analyzing such negotiation is to think about the negotiation process as an AI search algorithm where the outcome is characterized by decision variables to which the negotiation assigns values [Sathi and Fox, 1989; Sandholm, 1991; Conry *et al.*, 1991; Sycara *et al.*, 1991; Sandholm, 1993]. For example, in task allocation negotiation there is a decision variable for each task, and the variable's value is the name of the agent that the task is allocated to. There are several high-level families of peer-to-peer negotiation search algorithms. For example, the agents might negotiate one variable at a time, committing to the assignment before moving on to the next variable. This is analogous to constructive search in AI. As another example, the search might start from a *status quo* assignment of values to variables (for example, the initial assignment of tasks to agents before any negotiation begins), and then the agents might iteratively change the variable assignments whenever the agents relevant to the variables in question agree [Sandholm, 1991; 1993]. (For example, the current holder of a task can reallocate the task to another agent if they both agree.) This is analogous to iterative refinement search in AI.

2.1 Contracting based on marginal cost calculations

In the original contract net framework [Smith, 1980], agents allocated tasks among themselves. However, the framework was for cooperative agents only: an agent was assumed to take on a task whenever that was feasible. For self-interested agents, more sophisticated methods are needed. A key idea toward this direction was contracting based on marginal costs [Sandholm, 1991; 1993; 1996]. When a contract is proposed to an agent, the agent evaluates the cost of taking on the contract obligations (*e.g.*, tasks) by solving its local planning problem once with the new obligations and once without. The difference in the costs of those two local plans is the marginal cost of the obligations. An agent using this scheme accepts the proposal if the proposer pays it more than its marginal cost. Similarly, when proposing a contract, an agent computes its marginal value of unloading some obligations, and pays another agent up to that amount for the other agent to take on those obligations. A desirable engineering facet of this framework is that it separates the domain specific marginal cost calculator (planner) from the agent's domain independent negotiation module.

One practical consideration is that in many settings the local planning problems are intractable, so the marginal costs have to be approximated. For example, in the TRACONET system for automatically reallocating trucking tasks among dispatch centers, the local planning problems were $\mathcal{NP}$-complete vehicle routing problems with several side constraints. The TRACONET work included heuristic methods for deciding how carefully to approximate marginal costs [Sandholm, 1991; 1993; 1996; Sandholm and Lesser, 1995b]. This issue is revisited more formally in Section 5.

2.2 Iterative reallocation and combinatorial contracts

Another early idea in automated negotiation was to have the agents iteratively reallocate the items (tasks) [Sandholm, 1991; 1993]. An agent that had accepted a task could later contract out that task to some other agent, who in turn could contract it out, and so on. This proved to be highly effective in the TRACONET system. Marginal cost based contracting guarantees that every contract improves the utilities of the contract parties. Therefore, every agent's utility increases monotonically in the distributed negotiation that keeps reallocating tasks. It follows that the agents can enter and exit the negotiation dynamically without risking a loss.

The marginal cost based iterative reallocation negotiation can be viewed as a distributed hill-climbing search where the height on the hill is the sum of the agents' utilities [Sandholm, 1998]. Under this interpretation it is easy to show that contracting one item (task) at a time against a payment (aka. *original (O) contract*) does not generally lead to an optimal outcome: the search gets stuck in a local optimum. This can be addressed by combinatorial contracts that enlarge the neighborhood in the hill-climbing search: contracts where multiple items are exchanged for a payment (*cluster (C) contracts*),[1] where items from one agent are exchanged against items from another—potentially with a side payment—(*swap (S) contracts*), and where the contract can involve more than

[1]The TRACONET system was the first to use combinatorial bidding to allocate trucking tasks—an approach widely used commercially for procuring trucking services today.

two agents (*multiagent (M) contracts*). If all of these contract types are allowed in a single contract (*OCSM contract*), then agents that contract using the marginal cost principle will reach a global optimum (that is, a task allocation that maximizes *social welfare*, which is simply the sum of the agents' utilities) in a finite number of contracts—and no subset of those contract types suffices [Sandholm, 1998; 2000a]. Thus the agents can myopically make contracts in any order. From a hill-climbing search perspective, the neighborhood is large enough that from any allocation of tasks, a profitable contract exists that takes the agents to any other task allocation. Therein also lies a key weakness. Especially in a distributed setting, it can be difficult to find a combinatorial contract (involving multiple items and multiple agents) that will improve the current solution. Also, the sequence of hill-climbing contracts can be exponentially long in the worst case. Therefore, in practice, the optimal outcome is not found. Nevertheless, the combinatorial contracts help reach better outcomes than O-contracts [Andersson and Sandholm, 1999; 2000].

Another approach for avoiding local optima in search is backtracking. It turns out that a backtracking instrument can be constructed for negotiation as well, as the next section shows. It can easily be applied to the reallocation negotiation discussed in this section, instead—or in conjunction with—the combinatorial contract types [Andersson and Sandholm, 1998]. With that instrument, a backtracking option can be added to each (or only some) of the contracts. However, the backtracking instrument applies to basically any negotiation setting, and I will discuss it in that broader context.

2.3 Leveled commitment contracts to enable backtracking

Negotiating agents usually have to act under uncertainty, yielding behavior that is suboptimal in hindsight. For the purposes of the ensuing discussion, I divide the uncertainties faced by agents in negotiation into two high-level classes:

- **Domain uncertainty** stems from an agent not knowing how its local situation will change. Such changes affect the value (cost) and feasibility of the deals that the agent has made. For instance, which of an agent's resources will break (or become available) that affect the agent's costs—or even feasibility—of handling different combinations of its tasks?[2]
- **Negotiation process uncertainty** stems from an agent not knowing what future negotiation events will occur. Consider, for instance, the following uncertainties that an agent may face. Which of my pending bids will get accepted? Which (parts) of my tasks will I be able to subcontract out in the future, and at what prices? What tasks will I be offered and at what prices? The answers to all of these questions affect the cost of taking on (or letting go of) other obligation due to complementarity and substitutability: the cost of taking on an obligation usually depends on what other obligations one has.

[2] An additional source of uncertainty arises through the other agents' non-negotiation actions. Specifically, how will the others act regarding aspects that have not been contractually bound (this is pertinent in domains where those actions can affect the agent's utility by hindering or helping the agent)?

In automated negotiation systems for self-interested agents, contracts have traditionally been binding. They do not allow agents to accommodate future events that are uncertain due to domain uncertainty or negotiation process uncertainty. Both of these classes of uncertainty may also include subjective uncertainty due to an agent's limited capability to process information—for example computationally limited capability to do mental lookahead in a (sequential) negotiation process. If, in a negotiation, an agent has made a commitment that turns out unprofitable in hindsight, the agent would like to backtrack that commitment. For example, more lucrative offers can arrive later, or handling a task can turn out more costly than anticipated.

Backtracking in negotiation search can be enabled using an instrument called a *leveled commitment contract*, where each contract party can unilaterally decommit from the contract by paying a predetermined penalty [Sandholm and Lesser, 1995b; 2001; 2002]. This mitigates both domain uncertainty and negotiation process uncertainty, whether objective or subjective.[3] A concern with this is *strategic breach*: a rational self-interested agent is reluctant to decommit because there is a chance that the other party will decommit, in which case the former agent gets freed from the contract, does not have to pay a penalty, and collects a penalty from the breacher. (This is an example issue that arises due to self-interest, and renders inapplicable traditional backtracking techniques like distributed constraint satisfaction (*e.g.*, [Yokoo, 2000]) which assume that all parties execute the distributed algorithm faithfully.) Given the contract price, decommitting penalties, and the agents' prior distributions of the value of the contract, one can conduct a Nash equilibrium analysis of the decommitting game (in other words, one can find decommitting strategies for the agents such that each agent's strategy is a best response to the other's). Each agent's best-response strategy is defined by a threshold on the value of the contract for that agent. If the value is below that threshold, the agent will decommit. It turns out that strategic breach indeed occurs: an agent does not decommit when the contract's value drops below the point where paying the decommitting penalty is

[3] A practical type of subjective uncertainty stems from the fact that computing the value (cost) of taking on the obligations of a contract is often intractable—as discussed—so the agents have to resort to approximate marginal value calculation. Leveled commitment allows an agent to bid based on a rough value calculation. If the agent wins the bid, the agent can invest a more thorough value calculation. If the contract no longer looks beneficial in light of this more refined calculation, the agent can decommit. The fact that only the winning bidders carry out a refined calculation can save computation system wide. Also, the negotiations can be carried out faster because agents can bid based on less computation.

Leveled commitment can be used to increase the speed of negotiation in an additional way as well. An agent can make (low-commitment) offers to multiple recipients although those offers are mutually exclusive from the agent's perspective. In case more than one recipient accepts, the agent can backtrack from all but one. This allows the agent to address the recipients in parallel instead of addressing them one at a time and blocking to wait for an answer before addressing the next.

worth it; rather it needs to drop to a level further down before the agent decommits. However, despite such strategic breach, leveled commitment contracts improve the expected payoffs of all contract parties compared to any contract where backtracking is not an option [Sandholm and Lesser, 2001]. It follows that leveled commitment also enables contracts that would not be mutually beneficial without the backtracking option.

Leveled commitment contracts differ based on whether agents have to declare their decommitting decisions sequentially or simultaneously, and whether or not agents have to pay the penalties if both decommit. These mechanisms lead to different Nash equilibria. It is easy to see that in the sequential mechanisms the second mover never decommits if the first mover does; if the first mover does not, then the second mover will decommit if the value of the contract to him turns out to be so low that it is worth paying the penalty. In the simultaneous game where both pay the penalties if both decommit, as an agent's penalty approaches zero, the agent becomes truthful. On the contrary, in the simultaneous game where neither pays if both decommit, as an agent's penalty approaches zero, the agent does not become truthful but the other contract party does! Despite the fact that the equilibria of these mechanisms differ, surprisingly, among risk-neutral agents each of the mechanisms leads to the same expected payoffs to the agents if the contract price and decommitting penalties are optimized for each mechanism separately [Sandholm and Zhou, 2002]. For agents with risk attitudes, the different mechanisms yield a different sum of utilities, and the relative ranking of the mechanisms varies based on the exact utility functions.

Computing plays a key role in operationalizing the idea of leveled commitment. Given the contract price, the decommitting penalties, and piecewise linear prior distributions on the contract's value for the different contract parties, the Nash equilibrium decommitting thresholds for each mechanism can be computed in polynomial time in the number of pieces [Sandholm *et al.*, 1999]. Furthermore, given the piecewise linear priors, it turns out that the contract price and decommitting penalties that maximize the sum of the contract parties' payoffs can be determined in polynomial time in the number of pieces for each of the leveled commitment mechanisms. The reader is invited to try a leveled commitment contract optimizer prototype, *eCommitter*, on the web [Sandholm, 2002b]. Leveled commitment contracts and the algorithms also generalize to deals that involve more than two agents.

The theoretical results discussed above pertain to a single contract. In negotiation there is usually a web of contracts, and an agent's breach can cause the victim of the breach to want to breach on another contract, and so on. There is generally a tradeoff between allowing enough backtracking to sufficiently explore the space for a good outcome and not wasting time in deep cascades of decommits—or even infinite loops of decommitting and recommitting. That tradeoff can be controlled by carefully increasing the decommitting penalties over time [Andersson and Sandholm, 2001; 1998].

3 A paradigm shift to mediated clearing

The Achilles heel of peer-to-peer negotiation is negotiation process uncertainty. Agents make commitments without visibility into what is going to occur later on in the negotiation process (and what has already transpired in negotiation among other agents). Leveled commitment contracts reduce, but do not eliminate, the negative impact of such uncertainty. This uncertainty also introduces strategic problems. For example, if an agent expects a better deal in the future (which it cannot profitably handle if it takes on the contract currently offered to it), the agent may want to pass on the current offer. So, the agent is not best off acting myopically—not even in marginal cost based contracting or leveled commitment contracts as the basic analysis discussed in the previous section assumes. Rather, a rational agent would want to look ahead into the future, which in turn requires speculating what the other agents will do. Acting rationally equates to solving for the agent's best strategy in a game tree whose depth is at least the number of contracts that can occur in the system. Such lookahead is intractable in practice (although leveled commitment contracts have been studied in this context with a small number of tasks to allocate [Andersson and Sandholm, 2001]). Even if an agent can conduct such lookahead, uncertainty about the other agents' private information (their preferences, tasks, resources, *etc.*) causes the agent to make commitments that are suboptimal—in light of later negotiation events—for the agent and for social welfare [Andersson and Sandholm, 2001]. As discussed, there is an additional problem in peer-to-peer negotiation: it can be prohibitively complex to find a contract that improves the current solution, especially when using combinatorial contracts involving multiple items and multiple agents.

In light of these problems, it is clear that in many settings, economically better solutions can be obtained by collecting the agents' information to a mediated clearing point such as an auction server, and conducting a search (aka. a *clearing*) there to determine the outcome, rather than conducting a distributed "negotiation" search. The reason is that the mediated clearing has all the information in hand while the distributed search makes decisions based on incomplete views. Furthermore, the mediated clearing can be programmed to execute the search algorithm faithfully while in the distributed search the agents will act based on self-interest—potentially causing the search not to find an optimal outcome.[4] The mediated approach can be structured so that it motivates the agents to reveal their information truthfully; this will be discussed in Section 3.6. In short, the mediated approach removes the negotiation process uncertainty. (The domain uncertainty remains, and the leveled commitment methodology can be used to mitigate it.) In most electronic commerce applications the mediated approach also saves communication because there is many-to-one communication instead of many-to-many, and because each issue needs to be communicated only once—rather than repeatedly as is commonly the case in peer-to-peer

[4] In the distributed search it can even be difficult to identify when an optimal solution has been found, and special termination detection algorithms are usually needed [Sandholm, 1993; 1996; Walsh and Wellman, 2000].

negotiation.

The shift to mediated clearing introduces the need for technology for conducting the clearing. The following subsections describe the mediated approach in more detail, different forms of the clearing problem, and algorithms for solving it.

3.1 A canonical example: Combinatorial auction

Consider a setting where multiple distinguishable items (*e.g.*, a right shoe and a left shoe) are auctioned sequentially, and a bidder's valuation for a bundle of items is not the sum of those items' valuations. Most multi-item allocation settings exhibit such nonadditivity, for example, strategic sourcing, allocation of trucking lanes, electricity markets, as well as many task and resource allocation problems in computer science. To bid appropriately for the item that is auctioned first (right shoe), a bidder needs to guess which other items he will win in the later auction(s). This requires lookahead in a game tree, which is intractable if numerous items are auctioned. Even with exact lookahead, the bidder generally does not know deterministically what will transpire in the auction—due to incomplete information about the other bidders' valuations of the items. For example, some other bidder could be a collector of left shoes who also likes right shoes somewhat, but less than our bidder. This can cause our bidder to win just the right shoe that has no value to him, that is, a bundle of items that is undesirable to the bidder given the prices. While undesirable to our bidder, the outcome (that is, allocation of the items to the agents) also does not maximize social welfare. It would have been better to give both shoes to the other bidder. In general, the outcome might not maximize social welfare even if each bidder wins a bundle that is worth more than the bidder had to pay for it: an even better allocation of the items to the bidders might exist.

These problems, that are due to the negotiation process uncertainty of a sequential auction, can be overcome via a *combinatorial auction* where bids can be submitted on combinations (bundles) of items [Rassenti *et al.*, 1982]. For example, a bidder can say: "I am willing to pay up to $100 for items 2,3, and 7 together". This removes the need for lookahead and for speculation about others because the bidder can evaluate the value of item 2 *in the known context where the bidder hypothetically receives items 3 and 7 as well*. The bidder cannot get stuck with item 2 in an unprofitable way. This removal of the so called *exposure risk* makes bidding easier. It also causes the bidders to bid more aggressively because they do not have to factor in the potential downside of getting stuck with undesirable bundles; the aggressive bidding makes the seller better off. Finally, social welfare is maximized because the goods are allocated to the bidders that value them the most.

3.2 Substitutability and XOR-constraints

The model discussed above, and most other work on combinatorial auctions (see for example [Rothkopf *et al.*, 1998; DeMartini *et al.*, 1999]), are based on a setting where each bidder can bid on bundles of items, and any number of a bidder's bids can be accepted (except, of course, that bids on overlapping bundles cannot be accepted). This works well when bids are superadditive: $b_i(S \cup S') \geq b_i(S) + b_i(S')$, where b_i is the bid of agent i, and S and S' are disjoint sets of items. In other words, the current techniques focus on capturing synergies (complementarities) among items. However, in many auctions in practice, some items can at least partially substitute for others (*e.g.*, when bidding for an umbrella and a raincoat). For instance when bidding for landing slots for a given airplane flight, the bidder is willing to take any one of a host of slots, but getting more than one adds only slight value because extra slots beyond the first one obtained only serve as backup. Substitutability causes bids to be subadditive: $b_i(S \cup S') < b_i(S) + b_i(S')$. This can lead to problems. For example, what happens if agent i bids $b_i(\{1\}) = \$5$, $b_i(\{2\}) = \$4$, and $b_i(\{1,2\}) = \$7$, and there are no other bidders? The auctioneer could allocate items 1 and 2 to agent 1 separately, charging $\$5 + \$4 = \$9$ instead of $\$7$.

This problem can be removed by using a bidding language where the bidders can submit *XOR-bids*, that is, bids on bundles such that only one of the bids can get accepted [Sandholm, 2002a]. This allows the bidders to express general preferences with both complementarity and substitutability. In other words, the bidder can express any value function $v : 2^m \rightarrow \mathbb{R}$, where m is the number of items for sale in the auction. For example, a bidder in a 4-item auction may submit the following input to the auctioneer:

$$(\{1\}, \$4)\ \text{XOR}\ (\{2\}, \$4)\ \text{XOR}\ (\{3\}, \$2)\ \text{XOR}$$
$$(\{4\}, \$2)\ \text{XOR}\ (\{1,2\}, \$8)\ \text{XOR}\ (\{1,3\}, \$6)\ \text{XOR}$$
$$(\{1,4\}, \$6)\ \text{XOR}\ (\{2,3\}, \$6)\ \text{XOR}\ (\{2,4\}, \$6)\ \text{XOR}$$
$$(\{3,4\}, \$3)\ \text{XOR}\ (\{1,2,3\}, \$10)\ \text{XOR}$$
$$(\{1,2,4\}, \$10)\ \text{XOR}\ (\{1,3,4\}, \$7)\ \text{XOR}$$
$$(\{2,3,4\}, \$7)\ \text{XOR}\ (\{1,2,3,4\}, \$11)$$

While XOR-bids are fully expressive, representing one's preferences in that language often leads to large numbers of bids that are all combined with XOR. To maintain full expressiveness, but at the same time to make the representation more concise, one can use a bidding language called *OR-of-XORs* [Sandholm, 2002b; 2000b]. In this language, a set of bids can be combined with XOR, forming an *XOR-disjunct*. Multiple XOR-disjuncts can then be combined with non-exclusive ORs to represent independence (much like a lack of an edge represents independence in a Bayes net). For example, a bidder who wants to submit the same offer as in the example above, can do so by submitting the following more concise input to the auctioneer:

$$[(\{1\}, \$4)]$$
$$\text{OR}$$
$$[(\{2\}, \$4)]$$
$$\text{OR}$$
$$[(\{3\}, \$2)\ \text{XOR}\ (\{4\}, \$2)\ \text{XOR}\ (\{3,4\}, \$3)]$$

The XOR-bidding language is a special case of the OR-of-XORs language. Therefore, the shortest way to represent any particular value function in the OR-of-XORs language is never longer than in the simple XOR-bidding language. The reader is invited to try out XOR bidding and

OR-of-XORs bidding in our Internet auction server prototype, *eMediator* (see `http:// www.cs.cmu.edu/˜amem/eMediator`). Later other logical bidding languages were proposed, namely the XOR-of-ORs language where XORs combine OR-disjuncts, and the OR* language where XOR-constraints can be submitted between arbitrary pairs of bids [Nisan, 2000]. Recently, recursive logical bidding languages have also been proposed [Hoos and Boutilier, 2001; Boutilier, 2002].

3.3 Expressive competition beyond combinatorial auctions

One can view combinatorial auctions as an extremely special case of a broader approach that I call *expressive competition*. The vision is to empower market participants with potent expressiveness akin to human-to-human negotiation while at the same time harnessing the forces of competition (rather than 1-to-1 negotiation), the global scale of Internet auctions, and the speed and accuracy of algorithmic market clearing with all relevant information in hand.

Market types

There are three high-level market types for expressive competition, each of which can involve bidding on bundles, and expressions of substitutability (for example with XOR constraints using the OR-of-XORs language). In a *combinatorial auction*, there is one seller, and multiple buyers who bid. The *clearing problem* (aka. winner determination problem) is that of determining which bids win and which lose so as to maximize the sum of the winning bids' prices—under the constraint that every item can be allocated to at most one bid. In a *combinatorial reverse auction*, there is one buyer with a set of items he wants to procure, and multiple sellers who bid [Sandholm, 2002b; Sandholm *et al.*, 2002]. The clearing problem is that of determining which bids win and which lose so as to minimize the sum of the winning bids' prices—under the constraint that every item in the set gets procured. In a *combinatorial exchange* (aka. combinatorial double auctions) there are multiple buyers and multiple sellers [Sandholm, 2002b; Sandholm *et al.*, 2002; Walsh *et al.*, 2000]. A bidder can also act both as a buyer and as a seller, even in one bid. For example, one of his bids may state: "I want to buy a car, sell a boat, buy a bike, and get paid $150". There are two natural clearing objectives for exchanges [Sandholm and Suri, 2003]. In *surplus maximization*, the goal is to maximize the sum of the payments collected from winning bids that offer a payment, minus the sum of the payments paid to winning bids that require a payment. In *liquidity maximization*, the goal is to maximize the number or dollar volume of trades. In an exchange, the constraint on the clearing is that among the winning bids, supply meets demand on every item (if the exchange sells an item to someone, it also has to buy it from someone).

For each of these market types, there are two variants: the *single-unit* variant (described above), and the *multi-unit* variant. In the latter, there are multiple indistinguishable units of each distinguishable item in the market [Sandholm, 2002b; Sandholm and Suri, 2003; Leyton-Brown *et al.*, 2000b; Gonen and Lehmann, 2000; Sandholm *et al.*, 2002]. For example in a reverse auction, a buyer may want 200,000 nuts and 100,000 bolts. A bidder's bid may state: "I offer 50,000 nuts and 20,000 bolts for $2,500".

Finally, markets differ based on whether or not extra units can be thrown away for free. This is called *free disposal*. Most markets have free disposal. It makes the clearing problem easier because any solution where supply equals *or exceeds* demand is feasible. Without free disposal, supply needs to equal demand on every item.

Side constraints on the clearing: A form of extremely concise expressiveness

Expressive bidding enables the bidders to express their strengths and to avoid the exposure problem. Bidding on bundles (potentially with a language such as OR-of-XORs to enable substitutability to be expressed) is a simple form of expressive bidding, but representing one's preferences in such a language may require an expression of exponential length in the number of items. More concise forms of expressiveness can be used instead—or better, in addition. A key approach for accomplishing this is to allow bidders to express side constraints on the clearing [Sandholm and Suri, 2001b; Kalagnanam *et al.*, 2001]. For example in a reverse auction, a bidder could submit a number of bids (maybe on bundles, and potentially with XOR constraints), but in addition he could state that his capacity to produce tomatoes is only 60 tons. This would render infeasible all clearing solutions in which he is allocated more than 60 tons of tomato production. This is an instance of a class of constraints that I call *unit constraints*. Similarly, in an auction, a bidder could submit a number of bids (maybe on bundles, and potentially with XOR constraints), but in addition he could state that he has a budget constraint of $35,000,000. This is an instance of a class of constraints that I call *cost constraints*. Yet another form of concise expressiveness is to allow bidders to submit discount schedules, *e.g.*, "If I get to supply at least $9,000,000 of tomatoes to you, I will give you a 2% discount. At $15,000,000, I will raise the discount to 3%." Or the bidder could submit a supply (demand) curve where the per-unit price of tomatoes is a function of the quantity [Sandholm, 2002b; Sandholm and Suri, 2001a; 2002].

While work on combinatorial auctions has traditionally focused on increasing the expressiveness for the bidders, I view expressive competition as having two equally important parts, *expressive bidding* and *expressive bid taking*. A key approch toward expressive bid taking is to allow the bid taker to submit side constraints on the clearing. This allows him to model and honor legal constraints, for example the following cost constraint: "minority bidders have to win 10% of the auction". It also allows the bid taker to honor contractual obligation. For example, the buyer in a reverse auction for transportation services may have a contract that states that Joe's Trucking has to get at least $30,000,000 of business. This can be modeled as a side constraint in the clearing, and the clearing algorithm will decide exactly what services are procured from Joe's Trucking and what services from other providers. This approach allows one to reap the benefits of both long-term contracts and dynamic pricing—modes of trade traditionally considered mutually exclusive. Finally, the bid taker can submit his business rules as side constraints. He may use *counting constraints* such as "I don't want to deal with more than 200 winning suppliers, and I do not have the capability to handle more than 3 at my plant in Chicago". He may also use cost constraints like "I don't want any one supplier to win more than 15% of the business (so that my supply chain stays competitive for the long run)".

The use of side constraints allows one to find a market clearing solution that is *implementable* in the world because the real-world constraints are honored. By changing side constraint and running the clearing algorithm again, the bid taker can also conduct quantitative what-if analyses. For example, "How much would my procurement cost in this reverse auction increase if I decrease my supply base down to 17 suppliers?" Or, "How much more would I save in this reverse auction if I did not go with a sole-source contract for electrical supplies in my Cincinnati plant, but rather allowed 2 suppliers?"

In the 70 markets with expressive competition that we have fielded to date (see, *e.g.*, [CombineNet, Inc., 2003e; 2003c; 2003b; 2003d; 2003a]), we have seen hundreds of side constraint types. We abstracted them into seven general classes, the most prevalent of which are cost constraints, unit constraints, and counting constraints. This abstraction allows the clearing algorithm to be designed for a few classes rather than for hundreds of constraint types.

Non-price attributes in the clearing

An additional form of expressive competition stems from the fact that in many markets there are non-price attributes that are pertinent to the clearing problem, such as color, width, delivery date, quality, insurance terms, and so on. There are at least two reasons for introducing multi-attribute techniques into the clearing problem. First, in a basic combinatorial auction (or reverse auction or exchange), each item has to be completely specified. In many settings, this is overly restrictive. It is more desirable to leave some of the item attributes unspecified, so that each bidder can propose in his bids the attribute vector that is most suitable to him. Each bidder can also submit multiple bids with alternative attribute vectors, which is desirable because different attribute vectors are generally not equally valuable to the bid taker. Second, a bid from one bidder can be more valuable than the same bid from another bidder (due to bidder attributes such as historical data on timeliness and quality), and this should be taken into account in the clearing.

Multiattribute considerations can be integrated into combinatorial auctions and reverse auctions as follows [Sandholm and Suri, 2001b]. Let $\vec{a_j}$ be a vector of attributes. These can be item attributes and/or bidder attributes. Some of the attributes can be specific to one bid (say j) while others might not (such as quality of a certain line of products). The vector can include attributes revealed by the bidder as well as attributes whose values the bid taker gets from other sources such as historical performance databases. The bid prices can be re-weighted based on the additional attributes. The new price of any bid j is $p'_j = f(p_j, \vec{a_j})$, where p_j is the original price of that bid. The re-weighting function f is usually expressed by the bid taker (seller in an auction, buyer in a reverse auction)—either before or after the auction to characterize his preferences. (For example, the buyer of sea freight services may give a 4% advantage to bids that include less than 3 interim ports on the route from source to destination.) The clearing problem is then solved using these revised prices.

Unlike in auctions and reverse auctions where multiple attributes can be handled in a preprocessor to the clearing problem as shown above, in exchanges multiple attributes cannot be handled in a preprocessor. The reason is that in an exchange there are multiple bid takers (each buyer and each seller is a bid taker in this sense), and they may have different preference functions f over attribute vectors. Multiple attributes can still be handled, but their handling has to be incorporated into the clearing problem itself. This can be accomplished as follows. Treat items that have different values of the item attributes as different items. Then use a separate decision variable not just for each such item, but for each $\langle$ item, buyer, seller $\rangle$ tuple. This way each buyer (seller) can condition his bid price on the item attributes and on whom he is buying from (selling to). Conditioning on whom he is buying from (selling to) is pertinent when bidder attributes have to be taken into account.[5]

3.4 Complexity of the clearing problem

Expressive competition is a new way of conducting business, and has a host of advantages as discussed above. However, it requires solving the clearing problem—a combinatorial optimization problem. Many variants of it are hard, and the variants span an intriguing spectrum of worst-case complexity when it comes to the complexity of finding a feasible solution, an approximately optimal solution, or an optimal solution:

- As discussed, in the canonical combinatorial auction there is one unit of each item, bids can be submitted on bundles (and bids on overlapping bundles cannot both win), there is free disposal, and there are no XOR-constraints or other side constraints. Optimal clearing in this setting is $\mathcal{NP}$-complete [Rothkopf *et al.*, 1998].
- That problem is also inapproximable: no polynomial-time algorithm can guarantee a solution that is better than a bound $n^{1-\epsilon}$ from optimal, where n is the number of bids (unless P=ZPP) [Sandholm, 2002a].
- Optimal clearing in a combinatorial reverse auction and a combinatorial exchange is $\mathcal{NP}$-complete even if there is only one unit of each item, free disposal, and no XOR-constraints or other side constraints [Sandholm *et al.*, 2002].
- Combinatorial reverse auctions are not as inapproximable as combinatorial auctions: a bound $(1 + \ln M)$ can be obtained in polynomial time even in the multi-unit case, where M is the largest number of units that

[5] There also exist auctions and reverse auctions where the multi-attribute aspects cannot be handled in a preprocessor. This occurs if the bid taker's way of evaluating attributes depends on which bids win. In such cases, as in exchanges, the way in which the attributes are to be taken into account can be modeled in the clearing problem itself.

any bid contains [Sandholm *et al.*, 2002]. (This assumes the canonical setting where there is free disposal, and no XOR-constraints or other side cosntraints.)

- Finding a feasible solution is trivial in combinatorial auctions (accept no bids, or any one bid) and in combinatorial reverse auctions (accept all the bids; if this does not cover the demand, then nothing will). Without free disposal, even finding a feasible solution is $\mathcal{NP}$-complete in these variants, even in the single-unit case with no XOR-constraints or other constraints [Sandholm *et al.*, 2002] (this implies inapproximability to any ratio bound). The same results apply to combinatorial exchanges (the hardness applies if the trivial "no bids accepted" solution is excluded).
- XOR-constraints do not change the approximability of canonical combinatorial auctions [Sandholm *et al.*, 2002]. However, XOR-constraints make finding a feasible solution $\mathcal{NP}$-complete in combinatorial reverse auctions (even in the single-unit case with free disposal) [Sandholm *et al.*, 2002]. In other words, combinatorial reverse auctions are more approximable than combinatorial auctions, but this ordering reverses when XOR-constraints are introduced.
- Combinatorial exchanges inherit the inapproximability of both combinatorial auctions and combinatorial reverse auctions [Sandholm *et al.*, 2002].
- Cost constraints and unit constraints do not affect the complexity class of the clearing problem when bids can be submitted on bundles: the basic case where bids have to be accepted all or nothing remains $\mathcal{NP}$-complete, and the case where bids can be accepted fractionally can be solved in polynomial time using linear programming [Sandholm and Suri, 2001b]. However, XOR-constraints and other counting constraints make even the fractional case $\mathcal{NP}$-complete [Sandholm and Suri, 2001b]. There exist severe side constraints that restrict the search space enough so that even the case where bids have to be accepted all or nothing is optimally clearable in polynomial time [Sandholm and Suri, 2001b].
- If bids can be accepted fractionally, a combinatorial exchange (and auction and reverse auction) can be optimally cleared by accepting a very small number of bids fractionally: $m + 1$ if the objective is to maximize surplus, and $m + 2$ if the objective is to maximize liquidity [Kothari *et al.*, 2003]. Here m is the number of distinguishable *items* in the market. This clearing can be found using linear programming. However, if XOR-constraints are allowed, finding the surplus-maximizing clearing is $\mathcal{NP}$-complete even in the fractional case, even if there is just one item in the market with multiple units of it [Kothari *et al.*, 2003].
- In an auction where bids can be submitted on individual items only (not on bundles), winner determination is trivial: simply accept the highest bid on each item. Under budget constraints, optimal winner determination becomes $\mathcal{NP}$-complete. Curiously, if instead of a budget constraint each bidder submits a constraint on the number of items he wins, the auction can be optimally cleared in polynomial time using b-matching [Tennenholtz, 2000]. If bids can be accepted fractionally, winners can be determined in polynomial time using linear programming even with budget constraints. Under XOR-constraints (or other counting constraints), clearing is $\mathcal{NP}$-complete even if bids can be accepted fractionally [Sandholm and Suri, 2001b]. (If each bidder places an XOR-constraint between *every* pair of his bids—so that at most one of his bids can be accepted—then the problem becomes the *assignment problem* [Sandholm and Suri, 2001b]. The assignment problem can be solved in polynomial time [Kuhn, 1955].)
- Even if there is just one item in the market (multiple units of it), piecewise linear supply/demand curves are $\mathcal{NP}$-complete to clear optimally (in an exchange, auction, and reverse auction), but with linear supply/demand curves, optimal clearing can be done in polynomial time [Sandholm and Suri, 2001a; 2002].

In summary, there is a tradeoff between expressiveness (with its economic and usability advantages) and the computational complexity of clearing the market. While many variants of the clearing problem are hard in the worst case, in practice problems of real-world sizes can usually be solved. (This is most likely due to the co-evolution of clearing technology and expressive markets.) Experiments show that combinatorial reverse auctions tend to be easier to clear optimally than combinatorial auctions, which in turn are easier than combinatorial exchanges [Sandholm *et al.*, 2002].

3.5 Algorithms for the clearing problem

While the idea of a basic canonical combinatorial auction is two decades old, combinatorial auctions have traditionally not been used. The main reason is that the clearing problem is tough. In the last few years, hardware and especially clearing algorithms, have reached a level of scalability that enables combinatorial auctions of real-world sizes to be cleared optimally. As a consequence, numerous combinatorial auctions have emerged in industry.

The rest of this section focuses solely on clearing algorithms that find an optimal solution. Optimal clearing is important because real money is at stake, because approximate clearing yields solutions extremely far from optimal (at least in the worst case), because a large change in the winners can occur even if the solution changes slightly from optimum, and because suboptimal clearing ruins the incentive properties of the market (as will be discussed in the next subsection).

The first-generation special-purpose clearing algorithms for the canonical combinatorial auction used a search tree where branching was on items (Figure 1 left) [Sandholm, 2002a; 2000b; Fujishima *et al.*, 1999]. The newer, and significantly faster, clearing algorithms use a search tree where branching is on bids instead (Figure 1 right) [Sandholm and Suri, 2003; Sandholm *et al.*, 2001]. It has the advantage that there is more flexibility in variable ordering: at each branch point, commitment occurs on only one bid rather than on all of the bids that include a specific item. Another benefit is that, unlike the branch-on-items tree, the branch-on-bids tree easily supports all of the market designs for expressive competition discussed above [Sandholm and Suri, 2003; Sandholm *et al.*, 2002].

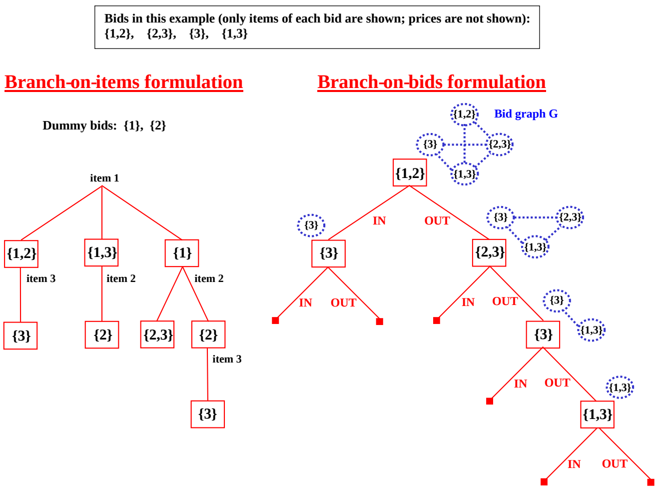

Figure 1: *Branching on items vs. branching on bids.*

Interestingly, although the problem is $\mathcal{NP}$-complete, both the branch-on-items tree [Sandholm, 2002a] and the branch-on-bids tree [Sandholm and Suri, 2003] are of polynomial size in the number of bids even in the worst case (and exponential in the number of items). This is desirable because the auctioneer usually controls the number of items for sale, but not the number of bids that happen to be submitted.

The techniques that make the branch-on-bids search fast in practice include bid ordering heuristics that dynamically choose the next bid to branch on, techniques for dynamically choosing the bid ordering heuristic itself based on context, upper bounding based on a linear programming relaxation of the remaining subproblem (and intelligent addition of cutting planes that reduce the size of the linear programming polytope, but do not cut out the optimal integer solution), lower bounding based on rounding techniques, decomposition of the problem when the bid graph (Figure 1 right) splits into independent components, techniques for enhancing upper and lower bounding based on information from sibling components, and methods for identifying and solving potential polynomially-solvable special cases at each node. In the interest of flow, I will not go further into those techniques here. Many of them are described in detail in specialized papers [Sandholm and Suri, 2003; Sandholm *et al.*, 2001].

Suffice it to say that the CABOB algorithm [Sandholm *et al.*, 2001] is, by and large, the fastest algorithm for the canonical combinatorial auction clearing problem currently, based on the widely adopted approach of evaluating clearing algorithms on standard randomly generated benchmark distributions [Sandholm, 2002a; Fujishima *et al.*, 1999; Leyton-Brown *et al.*, 2000a; Andersson *et al.*, 2000]. Interestingly, economically motivated random distributions (specifically the ones from CATS [Leyton-Brown *et al.*, 2000a]) tend to be very easy compared to totally random distributions [Sandholm *et al.*, 2001]. Similarly, real clearing problems are often easier than totally random ones. Our fielded algorithms systematically solve real reverse auctions with up to tens of thousands of items, hundreds of thousands of bids, and hundreds of thousands of side constraints to optimum.

3.6 Incentives to bid truthfully: The Vickrey-Clarke-Groves (VCG) mechanism

One concern is that bidders may bid insincerely. For example, in an auction where each winning bidder is charged the sum of the prices of his winning bids, the bidders are motivated to bid less than their true valuations of the items.

This issue can be overcome under the traditional assumption that each bidder i has a *quasilinear utility function:* $u_i(S, p_i) = v_i(S) - p_i$, where S is the bundle that bidder i wins, and p_i is the price that he has to pay. It turns out that under this assumption, a fully expressive bidding language combined with an optimal clearing algorithm is sufficient [Sandholm, 2002b; 2000b] and necessary [Nisan and Ronen, 2000] for being able to design auction mechanisms that yield a social welfare maximizing allocation and where every bidder's dominant strategy is to bid truthfully.[6] With such technology, bidding truthfully can be made a dominant strategy by using the *Vickrey-Clarke-Groves (VCG) mechanism* [Vickrey, 1961; Clarke, 1971; Groves, 1973]. This means that each bidder is motivated to bid truthfully regardless of how others bid—thus rendering speculation about others futile.[7]

The VCG mechanism can be applied to markets with expressive bidding as follows. The optimal clearing outcome is first computed as usual. The amount that an agent needs to pay is the sum of the others' winning bids had the agent not participated, minus the sum of the others' winning bids in the actual optimal outcome. So, the clearing problem has to be solved once overall, and once per winning agent without any of that agent's bids or side constraints.[8]

4 Preference elicitation in multiagent systems

Most, but not all, game-theoretic interaction mechanisms are so called *direct-revelation mechanisms*, where each participant reveals all of his private information (*e.g.*, bundle valuations in combinatorial auctions) completely up front. There

[6] Under extremely strong assumptions about the bidders' utility functions, truth-dominance can be accomplished with approximate clearing (see *e.g.*, [Lehmann *et al.*, 2002; Mu'alem and Nisan, 2002]), but because the clearing problem is inapproximable, the allocation is extremely far from optimal in the worst case.

[7] The VCG mechanism has also been used in mediated planning among software agents [Ephrati and Rosenschein, 1991; 1993; Ephrati, 1994].

[8] In auctions, the mechanism is budget balanced: the seller collects the nonnegative amounts paid by the bidders. In exchanges, an external benefactor is usually needed. In fact, with private valuation information on both the buy side and the sell side of the market, there is no mechanism for general quasilinear utility functions—even with just one unit to trade and only one buyer and one seller—that motivates the parties to participate in the mechanism, yields the social welfare maximizing outcome, and is budget balanced [Myerson and Satterthwaite, 1983].

are fundamental results that show that, in the absence of computation/communication limitations, this restriction comes at no loss in any sense (*cf.* the *revelation principle* [Mas-Colell *et al.*, 1995]). However, in practice such mechanisms are problematic because the agents may need to determine their own preferences via costly deliberation (*e.g.*, computing [Sandholm, 1993; 1996; 2000c; Larson and Sandholm, 2001b; 2001c]) or information gathering, and communicating complete preferences may be undesirable from the perspective of privacy or conserving bandwidth.

This issue can be addressed using a methodology where the mediator incrementally elicits the agents' private information on an as-needed basis in a manner directed to being able to determine the outcome (that would have come about had the agents revealed all of their private information to the mediator) [Conen and Sandholm, 2001]. It turns out that often the outcome can be determined while eliciting only a small portion of the agents' private information. A key insight is that in multiagent systems, what information is needed from a party depends on what information the other parties have revealed. This is a central motivation for interleaved preference elicitation from multiple agents. The approach offers the advantages of incremental problem solving (as in peer-to-peer negotiation) while reaping the benefits of mediated clearing discussed earlier. The rest of this section studies this approach as applied to combinatorial markets and voting.

4.1 Preference elicitation in combinatorial auctions

Preference elicitation is crucial in combinatorial markets because each agent has an exponential number of bundles to evaluate (and again, each such evaluation problem can be hard [Sandholm, 1993; Sandholm and Lesser, 1995b; Sandholm, 2000c; Larson and Sandholm, 2001b; 2001c]). Each agent would like to focus on a small number of bundles in order to minimize evaluation effort, communication, and loss of privacy. On the other hand, the social welfare among the agents, the seller's revenue, and the agent's utility will usually suffer if the agent's evaluation of a bundle that he would win is not communicated to the auctioneer. In a usual combinatorial auction it is difficult for the agent to decide which bundles to bid on because others' bids determine what bundles the agent would be competitive on.

To address this problem, we developed a methodology where an *elicitor* software, residing at the auctioneer, incrementally builds a model of the agents' valuation functions v_i [Conen and Sandholm, 2001]. The elicitor queries the agents about v_i, and fully assimilates the answers into its model. The next query to be asked is always chosen based on the answers so far. The elicitor in a sense opens up the clearing algorithm and elicits the inputs needed for determining the optimal outcome (allocation of items to agents). The agents are never asked for information that the elicitor can already infer from the answers, or that is known to be impertinent for determining the optimal allocation. The elicitor terminates the process when it has found a provably optimal allocation.

The rest of this section will focus on combinatorial auctions under the usual free disposal assumption, which almost always holds in practice. It gives structure to the elicitation problem because for each agent, the value of a bundle is no greater than the value of any superbundle of that bundle ($v_i(S) \leq v_i(S')$ whenever $S \subset S'$).

The general case

Even with free disposal, the worst-case communication complexity for even approximately optimally clearing the market is exponential in the number of items, regardless of the query types or the elicitation policy [Nisan and Segal, 2003]. The discussion below will focus on natural value queries: "what is your valuation of bundle S?" (Other query types can increase the efficiency of elicitation [Conen and Sandholm, 2001; Hudson and Sandholm, 2002; Conen and Sandholm, 2002b; 2002a].[9]) It turns out that in practice elicitation is very promising: only a vanishing fraction of all the queries are asked before the elicitor can clear the auction provably optimally [Hudson and Sandholm, 2002]! The preference elicitation methodology is also very promising in combinatorial reverse auctions [Hudson and Sandholm, 2003a] and combinatorial exchanges [Smith *et al.*, 2002].

One may also ask whether there exists a *universal revelation reducer* for combinatorial auctions, that is, a general elicitor algorithm that saves some elicitation (finishes finding an optimal outcome and proving its optimality without asking all value queries) on all instances where some instance-specifc elicitor saves some elicitation (that is, where the shortest certificate for verifying that a proposed outcome is optimal is shorter than the number of agents times the number of bundles). It turns out that a deterministic universal revelation reducer cannot exist, but randomized ones are easy to construct [Hudson and Sandholm, 2003b].

Restricted preferences for which the worst-case number of queries is polynomial in items

For restricted classes of v_i functions, the worst-case number of value queries needed is polynomial in the number of items being sold. Many of these classes are rich enough to exhibit both complementarity and substitutability. This subsection will present classes like that, which are also arguably natural for capturing valuation functions.

First, consider *read-once formulas*. A read-once formula is a function that can be represented as a tree, where the items for sale in the auction are at the leaves, together with the bidder's valuations for the individual items. The formula's output value is obtained by feeding in a bundle S of items to the leaves and reading the valuation $v_i(S)$ from the root. A leaf sends the item's valuation up the tree if the item is included in S, otherwise the leaf sends 0. Different types of gates can be used in the nodes of the tree. A SUM node sums the values of its inputs; a MAX node takes the maximum value of its inputs; an ALL node sums its inputs *unless* one of the inputs is zero, in which case

[9] Ascending combinatorial auctions (*e.g.*, [Bikhchandani *et al.*, 2001; Parkes and Ungar, 2000; Wurman and Wellman, 2000]) can be viewed as a special case of the preference elicitation framework where the queries are of the form: "Given these prices on items (and possibly also on bundles), which bundle would you prefer the most?".

the output is 0. For example, a legal function on 3 inputs might be $\mathrm{ALL}(2x_1, \mathrm{MAX}(10x_2, 4x_3))$, which gives value 12 to the bundle {1,2}, 6 to bundle {1,3}, and 0 to bundle {2,3}. Read-once formulas of this type allow for many natural preferences. For example, suppose items are flights and hotel rooms in different locations (*e.g.*, input $x_{i,j,0}$ represents the ith flight to location j, and $x_{i,j,1}$ represents the ith hotel room in location j) and we want to take just one trip. Then for each location j we could compute $\mathrm{ALL}(\mathrm{MAX}\{v_{i,j,0}x_{i,j,0}\}_i$, $\mathrm{MAX}\{v_{i,j,1}x_{i,j,1}\}_i)$, and then at the root of the tree we would take a MAX over the different destinations. More general gates are also possible. Let MAX_k output the sum of the k highest inputs, and $\mathrm{ATLEAST}_k$ output the sum of its inputs if there are at least k positive inputs, and 0 otherwise. Finally, let $\mathrm{GENERAL}_{k,l}$ be a parameterized gate capable of representing all the above types of gates. For instance, imagine that on a vacation to the Bahamas, Alice wanted entertainment. If she got to go out on at least three nights, then the trip would be worthwhile. Otherwise, she would rather stay home. Each night, she takes the maximum valued entertainment option. Then there is an $\mathrm{ATLEAST}_3$ node combining all of the different nights. In a different situation, imagine that Joe wants a more relaxing vacation in Hawaii, where he does not want to go out more than three nights. In this case, a MAX_3 gate will be useful. For each night, he chooses the best possible entertainment given to him. Then, he takes the best three nights of entertainment. Finally, imagine that Maggie wants a moderately active vacation, and is interested in going to Paris for a week, and wants at least three but no more than four nights of entertainment. Then a $\mathrm{GENERAL}_{3,4}$ gate will describe her preferences. It turns out that if the user's valuation function v_i can be expressed as a read-once formula with these gates (even if the user is not aware of that), then v_i can be elicited with a number of value queries that is polynomial in the number of items in the auction even in the worst case [Zinkevich *et al.*, 2003].[10] Furthermore, if a bidder's valuation function v_i is close to some read-once function with the gates discussed above, then a close model of v_i can be constructed with a polynomial number of value queries.

Second, consider the class of preferences that can be expressed as monotone polynomials. For example, $v_i(x) = ax_1x_2 + bx_2x_3x_4 + cx_3x_4$. This class is called *Toolbox DNF* because it captures settings where each agent has a set of tasks to accomplish (one per term in the polynomial), each task requiring a specific set of tools (the variables in the term) and each having its own value (the coefficient on that term). For example, the tools may be medical patents, and producing each medicine requires a specific set of patents. The value of a set of items to the agent is the sum of the values of the tasks that the agent can accomplish with those items. It turns out that any toolbox DNF valuation function v_i can be elicited in a polynomial number of value queries [Zinkevich *et al.*, 2003].

The power of interleaved preference elicitation from multiple parties

The above two example classes of valuation functions can be elicited efficiently even if there is only one party whose preferences are to be elicited. So, the efficiency does not derive from the fact that information from other agents restricts the amount of information needed. The next class, on the other hand, derives its ease of elicitation from this phenomenon.

Consider a combinatorial auction with two buyers that have their valuation functions in the form of the XOR bidding language discussed earlier, where each bidder can submit bids on bundles, and all of the bidder's bids are mutually exclusive. It turns out that if no bid includes more than $\log_2 m$ items (where m is the number of items in the auction), then the provably optimal allocation of items to the bidders can be determined in a worst-case polynomial number of value queries [Blum *et al.*, 2003].

Interestingly, there are classes of v_i functions where learning the functions requires an exponential number of value queries, while the provably optimal allocation can be constructed in a polynomial number of value queries [Blum *et al.*, 2003]. For other classes of v_i, being able to elicit enough to find the provably optimal allocation in a polynomial number of value queries implies that the v_i functions themselves can be learned in a polynomial number of value queries. So, sometimes there is an exponential benefit to interleaving the queries made to the different parties, while at other times the benefit between that and eliciting each agent's preferences separately is polynomially bounded.

Incentives to answer truthfully: *Ex post equilibrium* in an incremental push-pull multiagent elicitation mechanism

Motivating the bidders to answer queries truthfully is another key issue, and is exacerbated by the fact that the elicitor's queries leak information to the bidder about the answers that other bidders have given. Recently, a methodology was proposed by which elicitors can be made incentive compatible in the sense that every bidder (with a quasilinear utility function) answering the queries truthfully is an *ex post equilibrium* [Conen and Sandholm, 2001]. This means that bidding truthfully is each bidder's best strategy (for any prior probability distribution that he may hold about the other bidders) given that the other bidders bid truthfully. In other words, truthful bidding strategies form a (Bayesian) Nash equilibrium even in hindsight. (This does not mean that bidding truthfully is a dominant strategy; if some of the other agents bid insincerely and conditional on the elicitor's query stream to them, one may do better by bidding insincerely. In summary, implementation in *ex post* equilibrium is stronger than

[10] An analogous issue arises with shopping agents. Consider the following scenario. Alice goes to her software agent and asks it to help her purchase a vacation. In order to act on her behalf, the agent first needs to find out Alice's preferences (how much is a trip to Hawaii worth compared to a trip to the Bahamas, does it substantially increase the value to her if she can get some entertainment booked in advance, *etc.*). Then, after scouring the Internet, the agent needs to solve the computational problem of deciding on the best vacation package—the one that maximizes Alice's valuation minus the cost of the trip. In this scenario, there is no auctioneer. Rather, the elicitor is the buyer's helper. Again, the amount of querying can be prohibitively large when the buyer has general (monotone) preferences, but with these types of read-once preferences, Alice's valuation function can be determined in a polynomial number of value queries.

implementation in (Bayesian) Nash equilibrium, but weaker than implementation in dominant strategies.)

The methodology is the following. The mechanism is structured so that if all the bidders answer truthfully, the final allocation and payments follow the VCG mechanism. The amount a bidder has to pay is the sum of the others' revealed valuations for the bundles they get had the bidder not been given any of the items, minus the sum of the others' revealed valuations for the bundles they get in the actual optimal allocation. The elicitor can determine these payments by asking enough queries to be able to determine the welfare maximizing allocation overall, and *by asking extra queries to determine the welfare maximizing allocation for the auctions where each agent is ignored in turn*. The extra queries needed to determine the VCG payments are a negligible fraction of the queries needed to determine the optimal allocation in practice [Hudson and Sandholm, 2003a], and in some elicitation policies that information comes purely as a side effect with no extra queries at all [Conen and Sandholm, 2002b; 2002a].

Truthful answering is an *ex post* equilibrium even in elicitation mechanisms where the bidders are allowed to pass on some queries (as long as they answer enough queries to determine the optimal allocation and the VCG payments) and to answer queries that were never asked [Conen and Sandholm, 2001]. This yields a pull-push mechanism where the elicitor guides the preference revelation, but each bidder can also proactively reveal values on bundles on which it thinks it is competitive.

4.2 Preference elicitation in voting

Multiagent preference elicitation can also improve the efficiency of running elections [Conitzer and Sandholm, 2002d]. To determine the optimal outcome for a given voting protocol, it is generally not necessary to elicit complete preferences from all voters, and some voters' preferences may not need to be elicited at all. Selective preference elicitation increases privacy, and reduces the cost of voting (traveling to the voting site, spending time, *etc.*). Again, what, if any, information should be elicited from an agent depends on what other agents have revealed about their preferences so far.

However, it turns out that effective vote elicitation gives rise to challenging *computational* problems. In the Single Transferable Vote protocol (defined later), even knowing when enough has been elicited to determine the provably optimal outcome is $\mathcal{NP}$-complete, while this is easy for all other common voting protocols (defined later). Even for these protocols, determining whose votes to elicit is $\mathcal{NP}$-complete, even with perfect suspicions about how the agents will vote. (The exception is the plurality protocol—the most common voting protocol. There, everyone votes for one candidate and the candidate with the largest number of votes wins. In that protocol, effective elicitation is easy.) If the elicitor's suspicions are imperfect, then effective elicitation can even be $\mathcal{PSPACE}$-hard.

Elicitation can also introduce additional opportunities for strategic manipulation by the voters because the elicitor's queries leak information among the voters. In *coarse elicitation* (where each voter's entire ranking of candidates is elicited if the voter is queried at all) this can be avoided by making sure that the voter does not know how many voters have been queried before him. In *fine elicitation* (where a voter is asked pairwise preferences one pair of candidates at a time) this can be avoided by making sure that the voter does not know what/how many queries have been made to others, and by making sure that the *order* in which queries are made to this voter is fixed up front rather than dependent on others' answers.

5 Hard valuation problems

As discussed in the previous section, preference elicitation can significantly reduce the agents' effort of evaluating bundles in combinatorial auctions. However, it cannot eliminate the need to evaluate at least some bundles to some extent. In this section I present a deeper look into an agent's evaluation problem with an explicit model of computation, and illustrate new strategic issues that stem from it.

In many markets, even computing one's valuation for a single bundle (or individual item) is complex. For example when bidding for trucking lanes (*i.e.*, tasks), this involves solving two $\mathcal{NP}$-complete local planning problems: the vehicle routing problem with the new lanes of the bundle and the problem without them [Sandholm, 1993]. The difference in the costs of those two local plans is the cost (valuation) of taking on the new lanes.

However, in practice bidders (humans or their software agents) have limited computation and time, so they cannot exactly evaluate all, or even any, bundles—at least not without cost!

This leads to interesting incentive issues. For example, even in an auction where one object is being sold, should a bidder evaluate the object if there is a cost to doing so? According to traditional auction theory, truthful bidding is the dominant strategy in the celebrated Vickrey auction where the object is given to the highest bidder at the price of the second-highest bid [Vickrey, 1961] (this is the VCG mechanism, discussed earlier, applied to single-object auctions). However, it turns out that the Vickrey auction loses its dominant-strategy property if the bidder has the option to evaluate the object or not [Sandholm, 2000c]. Whether or not the bidder should pay the evaluation cost depends on the other bidders' valuations.

The issues run even deeper. If a bidder has the opportunity to approximate its valuation to different degrees, how much computing time should the bidder spend on refining its valuation? If there are multiple items for sale, how much computing time should the bidder allocate on different bundles? A bidder may even allocate some computing time to evaluate other bidders' valuations (*e.g.*, how much it would cost for a competing trucking company to take on a given set of lanes) so as to be able to bid more strategically; I call this *strategic computing*.

To answer these questions, we developed a deliberation control method called a *performance profile tree* for projecting how an anytime algorithm (a blackbox from the perspective of the deliberation controller) will change the valuation if additional computing is allocated toward refining (or improving) it [Larson and Sandholm, 2001c; 2001b; 2001a;

2002]. Unlike earlier deliberation control methods for anytime algorithms, the performance profile tree is a fully normative model of bounded rationality: it takes into account all the information that an agent can use to make its deliberation control decisions. (This is necessary in the game-theoretic context; otherwise a self-interested agent could take into account some information that the model does not.) Specifically, the projection of the anytime algorithm's performance is conditioned on the *path* of the run on the current problem instance, as well as static instance features.

Using this deliberation control method, the auction can be modeled as a game, where computing actions are part of the game. At every point, each agent can decide on which bundle to allocate its next step of computing as a function of the agent's computing results so far (and in open-cry auction format also the others' bids observed so far). At every point, the agent can also decide to submit bids. One can then solve this model for the (Bayesian) Nash equilibrium, where each agent's (deliberation and bidding) strategy is a best-response to the others' strategies. I call this a *deliberation equilibrium*. Table 1 shows in which settings strategic computing can and cannot occur in equilibrium. First, this depends on the auction mechanism.[11] Interestingly, this also depends on whether the agent has limited computing (such as a free desktop computer on which it can run until the auction's deadline) [Larson and Sandholm, 2001b] or costly computing (such as being able to buy any amount of supercomputing time where each cycle comes at a cost) [Larson and Sandholm, 2001c].[12]

	Auction mechanism	Speculation by perfectly rational agents?	Strategic computing? Limited computing	**Costly computing**
Single item	First price	yes	yes	yes
	Dutch	yes	yes	yes
	English	no	**no**	**yes**
	Vickrey	no	**no**	**yes**
Multiple items	First price	yes	yes	yes
	VCG	no	**yes**	**yes**

Table 1: *Does strategic computing occur? The most interesting results are in bold. As a benchmark from classical auction theory, the table also shows whether or not perfectly rational agents, that can determine their valuations instantly without cost, would benefit from considering each others' valuations when deciding how to bid.*

[11] The Vickrey and VCG mechanisms were discussed earlier. The first-price auctions are sealed-bid auctions where the winning bidders pay their winning bid prices. The Dutch auction is a descending-price auction where the first bidder gets the object at the current price. The English auction is an open-cry ascending auction where the highest bidder wins and pays the price of his bid.

[12] In these settings one can also determine how much such selfish computing hurts social welfare in the worst deliberation equilibrium [Larson and Sandholm, 2003].

6 Using computational complexity as a barrier to strategic manipulation

As the discussion of valuation computation in the previous section shows, agents' computational limitations can have adverse effects on the incentive properties of interaction mechanisms. This section demonstrates that the reverse can also be made to be true: one can use the fact that agents are computationally limited to achieve things that are not achievable via any mechanism among perfectly rational agents. In particular, I illustrate in a voting context that computational intractability can be used as a barrier to undesirable strategic behavior, thus circumventing a seminal economic impossibility result.

One key problem voting mechanisms are confronted with is that of *manipulation* by the voters. An agent is said to manipulate (vote strategically) when it does not rank the alternatives according to its true preferences, but rather so as to make the eventual outcome most favorable to itself. For example, if an agent prefers Nader to Gore to Bush, but knows that Nader has too few other supporters to win, while Gore and Bush are close to each other, the agent would be better off by declaring Gore as its top candidate. Manipulation is an undesirable phenomenon because collective choice schemes are tailored to aggregate preferences in a socially desirable way, and if the agents reveal their preferences insincerely, a socially undesirable candidate may be chosen.

The issue of strategic voting has been studied extensively. A seminal negative result, the *Gibbard-Satterthwaite theorem*, states that if there are three or more candidates, then in any nondictatorial voting scheme, there are candidate rankings of the other voters, and preferences of the agent under which the agent is better off voting strategically than sincerely [Gibbard, 1973; Satterthwaite, 1975]. (A voting scheme is called dictatorial if one of the voters dictates the outcome no matter how the others vote). So, a reasonable general nonmanipulable voting protocol does not exist! One approach around this impossibility is to construct desirable general nondictatorial voting protocols (under which manipulations exist by the impossibility theorem), but under which *finding a beneficial manipulation is prohibitively hard computationally.*

In order to discuss specific hardness results, I first review the most common protocols. In each protocol, each voter expresses his preferences as a linear order over candidates. The protocol then takes those expressions and imposes one of the candidates as the chosen outcome. In the protocols that are based on scores, the candidate with the highest score wins. In each of the listed protocols (even the ones that have multiple rounds), the voters submit their preferences up front. That is, the voters are not allowed to change their preference revelations during the execution of the protocol.

- *scoring protocols*. Let $\vec{\alpha} = \langle \alpha_1, \ldots, \alpha_c \rangle$ be a vector of integers such that $\alpha_1 \geq \alpha_2 \ldots \geq \alpha_c$. For each voter, a candidate receives α_1 points if it is ranked first by the voter, α_2 if it is ranked second *etc.* The *score* $s_{\vec{\alpha}}$ of a candidate is the total number of points the candidate receives. The *Borda* protocol is the scoring protocol with $\vec{\alpha} = \langle c-1, c-2, \ldots, 0 \rangle$. The *plurality* protocol (aka. majority rule) is the scoring protocol with $\vec{\alpha} = \langle 1, 0, \ldots, 0 \rangle$. The *veto* protocol is the scoring protocol with $\vec{\alpha} = \langle 1, 1, \ldots, 1, 0 \rangle$.

- *maximin* (aka. *Simpson*). For any two distinct candidates i and j, let $N(i, j)$ be the number of voters who prefer i to j. The *maximin score* of i is $s(i) = \min_{j \neq i} N(i, j)$.
- *Copeland*. For any two distinct candidates i and j, let $C(i, j) = +1$ if $N(i, j) > N(j, i)$ (in this case we say that i beats j in their pairwise election), $C(i, j) = 0$ if $N(i, j) = N(j, i)$ and $C(i, j) = -1$ if $N(i, j) < N(j, i)$. The *Copeland score* of candidate i is $s(i) = \sum_{j \neq i} C(i, j)$.
- *single transferable vote* (STV). The protocol proceeds through a series of $c - 1$ rounds. At each round, the candidate with the lowest plurality score (*i.e.*, the least number of voters ranking it first among the remaining candidates) is eliminated. The winner is the last remaining candidate.
- *plurality with run-off*. In this protocol, a first round eliminates all candidates except the two with the highest plurality scores. Then votes are transferred to these (as in the *STV* protocol). After that, a second round determines the winner among these two.
- *cup* (sequential binary comparisons). The cup is defined by a balanced binary tree T with one leaf per candidate, and an assignment of candidates to leaves (each leaf gets one candidate). Each non-leaf node is assigned the winner of the pairwise election of the node's children; the candidate assigned to the root wins. The cup protocol assumes that the assignment of candidates to leaves is known by the voters before they vote. In the *randomized cup* protocol [Conitzer and Sandholm, 2002b], the assignment of candidates to leaves is chosen uniformly at random after the voters have voted.

There are two natural alternative goals of manipulation. In *constructive manipulation*, the manipulator tries to find an order of candidates that he can reveal so that his favorite candidate wins. In *destructive manipulation*, the manipulator tries to find an order of candidates that he can reveal so that his hated candidate does not win. These are special cases of the utility-theoretic notion of improving one's utility, so the hardness results carry over to that setting.

6.1 Complexity of manipulation when the number of voters and the number of candidates grows

Unfortunately, finding a constructive manipulation is in $\mathcal{P}$ for the plurality, Borda, and maximin voting protocols [Bartholdi *et al.*, 1989], which are commonly used. The only voting protocol for which constructive manipulation is known to be $\mathcal{N}P$-hard is the STV protocol [Bartholdi and Orlin, 1991].[13]

However, by slightly tweaking the voting protocols that are easy to manipulate, they can be changed into ones that are hard to manipulate. In particular, we revise them so that before the original protocol is executed, one pairwise elimination round is executed among the candidates, and only the winning candidates survive to the original protocol (so, about half of the candidates are eliminated in the preround). This makes the protocols $\mathcal{N}P$-hard, $\#P$-hard, or $\mathcal{P}SPACE$-hard to manipulate constructively, depending on whether the schedule of the preround is determined before the votes are collected, after the votes are collected, or the scheduling and the vote collecting are carefully interleaved, respectively [Conitzer and Sandholm, 2003e]. We proved general sufficient conditions on voting protocols for this tweak to introduce the hardness, and showed that the plurality, Borda, maximin, and STV protocols satisfy those conditions. So, these commonly used voting protocols can be made hard to manipulate by simply using one elimination round.

6.2 Complexity of manipulation when the number of candidates is constant

All of the hardness results discussed above rely on both the number of voters and the number of candidates growing. The number of candidates can be large in some domains, for example when voting over task or resource allocations, but in many elections—such as presidential elections—the number of candidates is small. If the number of candidates is a constant, both constructive and destructive manipulation are in $\mathcal{P}$, regardless of the number of voters [Conitzer and Sandholm, 2002b]. This holds even if the voters are weighted, or if a coalition of voters tries to manipulate, but not both. When a coalition of weighted voters tries to manipulate, complexity can arise even for a constant number of candidates, as summarized in Tables 2 and 3 [Conitzer and Sandholm, 2002b; Conitzer *et al.*, 2003]. One lesson is that randomizing over instantiations of the mechanisms (such as schedules of a cup) can be used to make manipulation hard.

Number of candidates	2	3	4,5,6	≥ 7
Borda	$\mathcal{P}$	$\mathcal{N}P$-compl.	$\mathcal{N}P$-compl.	$\mathcal{N}P$-compl.
veto	$\mathcal{P}$	$\mathcal{N}P$-compl.	$\mathcal{N}P$-compl.	$\mathcal{N}P$-compl.
STV	$\mathcal{P}$	$\mathcal{N}P$-compl.	$\mathcal{N}P$-compl.	$\mathcal{N}P$-compl.
plurality with runoff	$\mathcal{P}$	$\mathcal{N}P$-compl.	$\mathcal{N}P$-compl.	$\mathcal{N}P$-compl.
Copeland	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{N}P$-compl.	$\mathcal{N}P$-compl.
maximin	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{N}P$-compl.	$\mathcal{N}P$-compl.
randomized cup	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{N}P$-compl.
cup	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$
plurality	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$

Table 2: *Complexity of constructive weighted coalitional manipulation.*

Number of candidates	2	≥ 3
STV	$\mathcal{P}$	$\mathcal{N}P$-compl.
plurality with runoff	$\mathcal{P}$	$\mathcal{N}P$-compl.
randomized cup	$\mathcal{P}$	?
Borda	$\mathcal{P}$	$\mathcal{P}$
veto	$\mathcal{P}$	$\mathcal{P}$
Copeland	$\mathcal{P}$	$\mathcal{P}$
maximin	$\mathcal{P}$	$\mathcal{P}$
cup	$\mathcal{P}$	$\mathcal{P}$
plurality	$\mathcal{P}$	$\mathcal{P}$

Table 3: *Complexity of destructive weighted coalitional manipulation.*

[13] It is $\mathcal{N}P$-hard also for the *Second Order Copeland* protocol [Bartholdi *et al.*, 1989], but the hardness is driven solely by the tie-breaking rule.

All of the hardness results discussed above hold even if the manipulators know the nonmanipulators' votes exactly. Under weak assumptions, if weighted coalitional manipulation with complete information about the others' votes is hard in some voting protocol, then individual and unweighted manipulation is hard when there is uncertainty about the others' votes [Conitzer and Sandholm, 2002b].

Computation not only serves as a means to circumventing incentive problems as dicussed above, but it can also serve as the means for designing appropriate incentives as discussed in the next section.

7 Automated mechanism design

The aggregation of conflicting preferences for choosing an outcome is a central problem in multiagent systems, be the agents humans or software. The key difficulty is that the agents may report their preferences insincerely (*i.e.*, manipulate, as we just discussed in a voting setting). *Mechanism design* is the art of designing the rules of the game so that the agents are motivated to report their preferences truthfully and a desirable outcome is chosen.[14] The desirability objective can be, for example, social welfare, seller's revenue, fairness, or some tradeoff among these.

Mechanism design has traditionally been a manual endeavor. The designer uses experience and intuition to hypothesize that a certain rule set is desirable in some ways, and then tries to prove that this is the case. Alternatively, the designer formulates the mechanism design problem mathematically and characterizes desirable mechanisms analytically in that framework. These approaches have yielded a small number of canonical mechanisms over the last 40 years, each of which is designed for a class of settings and a specific objective. For example, the *VCG* and *dAGVA* [d'Aspremont and Gérard-Varet, 1979; Arrow, 1979] maximize social welfare among the agents in the class of settings where the agents have quasilinear utility functions. Mechanism design research has also yielded impossibility results that state that no mechanism works across a class of settings (for varying definitions of "works" and varying classes). For example, the Gibbard-Satterthwaite theorem discussed in the previous section states that for the class of general preferences, no mechanism works in the sense that 1) the mechanism's outcome can be any one of at least three candidates, 2) the mechanism is nondictatorial, and 3) every agent's dominant strategy is to reveal his preferences truthfully.

In sharp contrast to manual mechanism design, I envision a systematic approach where the mechanism is automatically created for the setting and objective at hand. This has several advantages. First, it can be used even in settings that do not satisfy the assumptions of the classical mechanisms. Second, it may allow one to circumvent the impossibility results: when the mechanism is designed to the setting (instance) at hand, it does not matter that it would not work on preferences beyond those in that setting (*e.g.*, for a class of settings). Even when the optimal mechanism—created automatically—does not circumvent the impossibility, it always minimizes the pain entailed by impossibility. Third, it may yield better mechanisms (in terms of stronger nonmanipulability guarantees and/or better outcomes) than the canonical mechanisms because the mechanism capitalizes on the particulars of the setting (the probabilistic information that the mechanism designer has about the agents' preferences). Given the vast amount of information that parties have about each other today, it is astonishing that the canonical mechanisms (such as first-price reverse auctions), which largely ignore that information, have prevailed thus far. I foresee an imminent revolution, where future mechanisms will be created automatically. For example, imagine a Fortune 1000 company automatically creating its procurement mechanism based on its statistical knowledge about its suppliers (and potentially also the public prices of the suppliers' inputs, *etc.*). I call this vision *automated mechanism design* [Conitzer and Sandholm, 2002c].[15]

7.1 The computational problem

As a first step toward fulfilling this vision, we modeled mechanism design as an optimization problem, and studied its complexity. In the model, each agent can have any one of a finite number of utility functions. An agent's utility function is private information. The mechanism designer has a prior probability distribution over each agent's possible utility functions. The first constraint to the problem (the *incentive compatibility constraint*) is that each agent has to be motivated to reveal its utility function truthfully regardless of what utility function the agent has. This comes in two variants. In the first (called *dominant strategy implementation*), the agent has to be no worse off by revealing his true utility function regardless of what utility functions the other agents reveal. In the second, the agent has to be no worse off, *in expectation*, by revealing his true utility function. (The expectation is taken as a weighted average over the possible truthful utility function revelations of the other agents). The second constraint to the problem (the *participation constraint*) is that each agent has to be no worse off by participating in the mechanism than not participating (otherwise a rational agent would not participate). This again comes in two variants. In the first, the agent has to be no worse off regardless of what utility functions the other agents reveal. In the second, the agent has to be no worse off *in expectation*. The input to the optimization also includes the designer's objective. The output is a mapping from utility function revelations to outcomes (or in the case of randomized mechanisms, to probability distributions over outcomes).

In settings without side payments, such as voting, designing an optimal (*e.g.*, expected social welfare maximizing) deterministic mechanism is $\mathcal{NP}$-complete [Conitzer and Sandholm, 2002c].[16] If side payments are allowed, designing a

[14] A central result in game theory, the *revelation principle*, allows the designer to restrict attention to such truthful mechanisms without loss in the objective [Mas-Colell *et al.*, 1995].

[15] Note that automated mechanism design is completely different from so called *algorithmic mechanism design* [Nisan and Ronen, 2001]. In the latter, the mechanism is designed manually with the goal that *executing* the mechanism is computationally tractable. On the other hand, in automated mechanism design, the mechanism itself is designed automatically.

[16] This actually holds for *any* solution concept from noncoopera-

deterministic mechanism is easy if the designer's objective is social welfare, but $\mathcal{NP}$-complete more generally (for example, if the objective is to maximize the expected revenue collected from the bidders—as is the objective in some auctions) [Conitzer and Sandholm, 2003b]. Interestingly, if one allows randomized mechanisms, the mechanism design problem becomes solvable in polynomial time using linear programming.[17] In other words, the designer can tackle the computational complexity, introduced by its uncertainty about the agents, by making the agents face additional uncertainty. This comes at no loss, and in some cases at a gain, in the mechanism designer's objective.

If the agents' utility functions are additively decomposable into independent issues, the input to automated mechanism design can be represented (potentially exponentially) more concisely. In that representation it is $\mathcal{NP}$-complete (even under strong restrictions) to design a mechanism that maximizes one of the following objectives: 1) expected social welfare when payments are not possible, 2) a general objective function even when payments are possible, and 3) expected revenue collected from the agents [Conitzer and Sandholm, 2003c]. Again, a randomized mechanism can be designed in polynomial time. So, the complexity as a function of the input length is the same in the concise representation as it is in the flat representation. In other words, due to its potentially exponentially shorter input length, the structured representation allows potentially exponentially faster automated mechanism design.

7.2 Applications

In initial experiments, automated mechanism design produced the following highlights [Conitzer and Sandholm, 2003a]:

- It reinvented the celebrated Myerson auction [Myerson, 1981], which maximizes the seller's expected revenue in a 1-object auction.
- It created expected revenue maximizing combinatorial auctions. This has been a long-standing recognized open research problem in (manual) mechanism design [Avery and Hendershott, 2000; Vohra, 2001]. The general form for such an auction is still unknown, but automated mechanism design created prior-specific optimal mechanisms. (In the manual mechanism design literature, even the problem with only two objects for sale is open; only a case with very special form of complementarity and no substitutability has been solved [Armstrong, 2000].)
- It created optimal mechanisms for divorce settlements, both with a benevolent arbitrator that tries to maximize the sum of the divorcees' utilities (with and without side payments), and an arbitrator that tries to maximize revenues collected from the divorcees.
- It created optimal mechanisms for a public good problem (deciding whether or not to build a bridge). The VCG mechanism could be used in this setting as long as each agent's utility function is quasilinear. However, in the VCG mechanism, nonnegative payments are collected from the voters (intuitively, the payments are collected in order to avoid the free rider problem), and those payments have to be burned. According to a seminal impossibility result, this problem plagues *any* mechanism that applies to general quasilinear utility functions, yields a social welfare maximizing decision, and makes truthful reporting of utility functions a dominant strategy [Green and Laffont, 1979]. The automated mechanism design approach allowed us to incorporate money burning as a loss in the social welfare objective, and maximize that revised objective. We had automated mechanism design create an optimal mechanism for the bridge building scenario under each variant of the incentive compatibility constraint discussed above (with the deterministic participation constraint). In neither variant was money ever burned. In the probabilistic variant, the bridge was always built if and only if that was best for the agents. (In the deterministic variant this was not always the case.) For the probabilistic variant of incentive compatibility, the general-purpose *dAGVA* mechanism could be used to yield the social welfare maximizing choice without burning money [d'Aspremont and Gérard-Varet, 1979; Arrow, 1979]. However, a seminal economic impossibility result shows that no mechanism for general quasilinear utility functions yields the social welfare maximizing choice, maintains budget balance, *and satisfies the participation constraint (even the probabilistic variant)* [Myerson and Satterthwaite, 1983]. As the experiment above showed, automated mechanism design can circumvent this impossibility. It constructed a mechanism that satisfies all these desiderata, and actually the *deterministic* (*i.e.*, stronger) variant of the participation constraint.
- It created optimal mechanisms for public goods problems with multiple goods. This is the public goods analog of combinatorial auctions.

tive game theory [Conitzer and Sandholm, 2002a], not just the ones discussed through the constraints above. All of the hardness results discussed in this section hold even with just 2 agents.

[17] This holds for any mechanism design objective that is linear in the outcome probabilities.

8 Safe exchange mechanism design

Mechanism design is not only needed for deciding on an outcome among agents. It is also key for *executing* the outcome. For example, if the outcome is a joint plan, how should it be executed so that no agent is motivated to deviate along the way [Braynov, 1994; Braynov and Sandholm, 1999]? From the perspective of markets, an important type of joint plan is the plan for exchanging items and payments between parties. Nondelivery is a major problem in exchanges, especially in electronic commerce: the supplier might not deliver the goods or the demander might not pay. A recent study shows that 6% of consumers with on-line shopping experience reported products or services that were paid for, but never received [National Consumers League, 1999].

8.1 Deal chunking

In some settings, mechanism design can be used to enable safe exchanges without legal enforcement or escrow companies. One such approach is our methodology where the exchange is split into chunks which the agents deliver in alternation [Sandholm, 1996; 1997].[18] The mechanism is practical when such splitting incurs little cost, as is the case with digital goods, computation time, many web services, and many investment instruments, for instance. *eExchangeHouse*, a safe exchange planner prototype, automatically determines a safe exchange plan for the exchange setting at hand such that neither party has incentive to vanish before completing the exchange [Sandholm and Ferrandon, 2000]. Only some ways of splitting the exchange into chunks and some sequences of delivering the chunks are safe in this sense [Sandholm, 1996; Sandholm and Lesser, 1995a]. The planner's algorithms for chunking and chunk sequencing provably find a safe exchange plan if one exists, and determine the shortest safe plan. The algorithms, as well as the amount of input that is solicited from the users, vary based on whether the exchanged items and units of each item are dependent or independent in terms of their value to the exchange parties.

8.2 A general model for safe exchange

In order to more broadly study the possibilities of using mechanism design to enable safe exchange, we developed a unified model of exchange mechanisms [Sandholm and Wang, 2002]. A key idea behind the model is that at any point of the exchange, each agent has a (potentially empty) set of items that he possesses, and a (potentially empty) *allocation set* that includes the items that he can reallocate—except not to himself—or destroy. The two sets need not be the same. An item can simultaneously be in one agent's possession set and in another agent's allocation set. Other aspects of the model include transfer costs and defection costs (how much of a reputation loss or risk would an agent face if he defaulted before completing the exchange).

The model captures the disparate earlier safe exchange approaches such as cryptographic coin ripping [Jakobsson, 1995], digital signatures, and our game-theoretic chunking mechanism discussed above. It also allows one to creatively and systematically think about, and analyze, novel exchange mechanisms. For example, our *reputation locking* mechanism stemmed from this model. It works as follows: 1) agent A allows agent B to encrypt A's reputation in the public database (*e.g.*, eBay), 2) agent B delivers to A, 3) agent A delivers to B, and 4) B decrypts A's reputation back into plaintext. In this mechanism, A's reputation is used as a virtual item that is is temporarily transferred into B's allocation set. If A does not deliver, B does not give back the reputation.

Being an overarching framework, the model also allows one to study what is inherently possible and impossible in safe exchange (with and without a trusted third party, and with an offline third party that only gets involved if the exchange fails).

9 Conclusions and perspective

Collective choice settings are ubiquitous and important, whether the agents are humans or software. Game theory provides a basis for engineering the incentives into the interaction mechanism so that a desirable outcome is chosen even though every agent acts based on self-interest. However, a host of computer science issues not traditionally addressed in game theory have to be addressed in order to make mechanisms work in the real world. Those computing, communication, and privacy issues are deeply intertwined with the economic incentive issues, as this writeup has illustrated.

Here, I would like to draw some high-level conclusions from the results presented above. Peer-to-peer negotiation suffers from negotiation process uncertainties that can be eliminated by using a mediator, such as an auction server, that collects the agents' private information and runs a clearing algorithm on that data to determine the outcome.[19] Domain uncertainty remains even in this approach, and *leveled commitment contracts* can be used to mitigate it to the economic benefit of all contract parties.[20] *Expressive competition* is a new form of interaction that empowers market participants with potent expressiveness akin to human-to-human negotiation (this has economic advantages and makes bidding easier) while at the same time harnessing the forces of competition, the global scale of the Internet, and the accuracy of market clearing with all relevant information in hand. The mediated approach with optimal clearing is feasible at a grand scale even with expressive competition, despite the fact that most variants of the clearing problem are hard and inapproximable in the worst case. The economic and computational efficiencies that mediated expressive competition offers suggest that in the future, marketplaces will merge into larger ones. To a certain extent this trend is already underway. For example, in the last couple of years large corporations have undergone a massive transition from plant-based procurement to global procurement.

The mediated approach can be made to require dramatically less information from the agents—especially in combinatorial markets—by using selective incremental multiagent preference elicitation. This decreases the agents' valuation determination costs (and other preference determination efforts) and communication costs. It also enhances privacy. It can be made into an incentive-compatible push-pull mechanism where the information revelation is guided by both the elicitor (auctioneer) and each agent. This makes sense be-

[18] Similar protocols have been recently studied by others, *e.g.*, [Matsubara and Yokoo, 2000].

[19] The clearing algorithm—usually a specialized tree search—can run on multiple machines. This is usually the case in large-scale applications, but the machines are usually co-located rather than at each agent's location. Also, there is no notion of one machine representing one agent. So, the distribution that is motivated by computational efficiency does not correspond to the distribution of agents (that is, the distribution of self-interest and private information).

[20] Leveled commitment can also mitigate the negotiation process uncertainty that arises if an agent participates in negotiations mediated by different mediators, and the negotiations' outcomes are not independent from the perspective of the agent's valuation. In that sense, leveled commitment can serve as part of the "glue" needed between marketplaces. Furthermore, leveled commitment can be used to mitigate the uncertainty that arises if one mediator runs multiple clearings over time, for example at fixed intervals or every time a new bid arrives. (For online algorithms for market clearing, see [Blum *et al.*, 2002].)

cause each agent has private information that suggests what should be revealed by the agent, and the auctioneer accrues information about the other agents that affects what information from the agent is pertinent.

A deeper look into an agent's evaluation problem shows that valuation determination costs ruin the incentives of classical auction mechanisms, and give rise to a new phenomenon I call *strategic computing*: using one's limited computing to approximate others' preferences at the cost of approximating one's own. Whether strategic computing occurs depends on the auction mechanism and the type of computational constraint (costly computing vs. limited computing).

While computational constraints can cause strategic problems, the reverse can also be made to be the case. Computational hardness can be used as the barrier to manipulation. This is especialy desirable in settings where economic mechanism design (incentive engineering) is known to fall short.

Computing can also be used to automatically design the mechanism, for instance the rules of a divorce settlement, auction, or public goods problem. Because the mechanism is designed for the specific setting at hand (objective and information about the agents), it often yields a better mechanism than the ones known to date. It can also circumvent seminal impossibility results.

Carefully designed mechanisms are needed not only for choosing an outcome, but also for executing the outcome, such as a joint plan. I illustrated this in the context of designing safe exchange mechanisms for anonymous parties in Internet commerce.

10 Future research directions

This is a fertile and relatively new area where AI and computer science at large can have enormous theoretical and practical impact. Work on expressive competition includes the ongoing quest for faster clearing algorithms and the pursuit of concise—perhaps application-specific—forms of expressive bidding that are easy to use in bidding and elicitation, and fast to process by a clearing algorithm. Automated mechanism design is still in its infancy, and holds significant promise for future research. It should be tested on new classes of problems, and applied to real-world settings. It could also be applied to design safe exchange mechanisms (using the general safe exchange model that we introduced)—an endeavor that has traditionally taken considerable creativity, yet has not delivered mechanisms (or impossibility results) for all settings. Known game-theoretic results that characterize properties of mechanisms could be used to reduce the search space in automated mechanism design. Furthermore, automated mechanism design could be used on a variety of settings (real or artificially generated) to see whether new canonical mechanisms and mechanism design principles can be inferred.

Another significant research stream arises from the observation that mechanism design should take into account the agents' computational constraints. Our *performance profile tree* based deliberation control method together with the idea of *deliberation equilibrium* provide a normative model of bounded rationality in multiagent systems, which is needed to determine how computationally constrained self-interested agents would behave in a given mechanism. This allows one to evaluate mechanisms for computationally constrained agents, and hopefully paves the way to designing such mechanisms. Not only auction and voting mechanisms, but also multiagent preference elicitation mechanisms should be designed with this methodology. This methodology could also be used to design mechanisms that are computationally hard to manipulate, where hardness is measured not in terms of worst-case complexity, but informed by game-theoretic deliberation control. (Other approaches to improving upon worst-case measures include designing mechanisms where manipulation is average-case hard, or even hard on every—carefully constructed—instance such as in factoring.) This methodology could even yield new mechanism design principles. As discussed, the central design principle in mechanism design, the *revelation principle*, ceases to meaningfully hold under computational or communication constraints. In such settings it can be better to use multi-stage mechanisms such as preference eliciation, unlike the principle suggests. Also, in such settings it has been theoretically demonstrated that there is some benefit to allowing for mechanisms where insincere preference revelation occurs [Conitzer and Sandholm, 2003d], unlike the principle suggests. Is there a significant practical benefit to be gained from such mechanisms? What would such mechanisms look like? Are there principles for constructing them?

Finally, are there unforeseen novel ways—beyond enabling expressive competition, multiagent preference elicitation, and automated mechanism design—of using computing to enhance collective choice?

References

[Andersson and Sandholm, 1998] Martin Andersson and Tuomas Sandholm. Leveled commitment contracting among myopic individually rational agents. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS)*, pages 26–33, Paris, France, July 1998.

[Andersson and Sandholm, 1999] Martin Andersson and Tuomas Sandholm. Time-quality tradeoffs in reallocative negotiation with combinatorial contract types. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 3–10, Orlando, FL, 1999.

[Andersson and Sandholm, 2000] Martin Andersson and Tuomas Sandholm. Contract type sequencing for reallocative negotiation. In *Proceedings of the Twentieth International Conference on Distributed Computing Systems*, Taipei, Taiwan, April 2000.

[Andersson and Sandholm, 2001] Martin Andersson and Tuomas Sandholm. Leveled commitment contracts with myopic and strategic agents. *Journal of Economic Dynamics and Control*, 25:615–640, 2001. Special Issue on Agent-Based Computational Economics. Early version: National Conference on Artificial Intelligence (AAAI), p. 38–45, Madison, WI, 1998.

[Andersson *et al.*, 2000] Arne Andersson, Mattias Tenhunen, and Fredrik Ygge. Integer programming for combinatorial auction winner determination. In *Proceedings of* *the Fourth International Conference on Multi-Agent Systems (ICMAS)*, pages 39–46, Boston, MA, 2000.

[Armstrong, 2000] Mark Armstrong. Optimal multi-object auctions. *Review of Economic Studies*, 67:455–481, 2000.

[Arrow, 1979] Kenneth Arrow. The property rights doctrine and demand revelation under incomplete information. In M Boskin, editor, *Economics and human welfare*. New York Academic Press, 1979.

[Avery and Hendershott, 2000] Christopher Avery and Terrence Hendershott. Bundling and optimal auctions of multiple products. *Review of Economic Studies*, 67:483–497, 2000.

[Bartholdi and Orlin, 1991] John J. Bartholdi, III and James B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[Bartholdi *et al.*, 1989] John J. Bartholdi, III, Craig A. Tovey, and Michael A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[Bikhchandani *et al.*, 2001] Sushil Bikhchandani, Sven de Vries, James Schummer, and Rakesh V. Vohra. Linear programming and Vickrey auctions, 2001. Draft.

[Blum *et al.*, 2002] Avrim Blum, Tuomas Sandholm, and Martin Zinkevitch. Online algorithms for market clearing. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 971–980, San Francisco, 2002.

[Blum *et al.*, 2003] Avrim Blum, Jeffrey Jackson, Tuomas Sandholm, and Martin Zinkevich. Preference elicitation and query learning, 2003.

[Boutilier, 2002] Craig Boutilier. Solving concisely expressed combinatorial auction problems. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 359–366, Edmonton, Canada, 2002.

[Braynov and Sandholm, 1999] Sviatoslav Braynov and Tuomas Sandholm. Power, dependence and stability in multiagent plans. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 11–16, Orlando, FL, July 1999.

[Braynov, 1994] Sviatoslav Braynov. Deviation-proof plans in open multiagent environments. In *Proceedings of the 11th European Conference on Artificial Intelligence*, pages 274–278, August 1994.

[Clarke, 1971] E H Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[CombineNet, Inc., 2003a] CombineNet, Inc. CombineNet improves negotiations, streamlines sea freight allocations, for international technology and research corporation, 2003. Case study white paper at www.CombineNet.com/CUSTOMERS/case_histories/ seafreight_fortune100chemicalco.pdf.

[CombineNet, Inc., 2003b] CombineNet, Inc. CombineNet interoperability enhances sourcing systems for international food manufacturer, 2003. Case study white paper at www.CombineNet.com/CUSTOMERS/case_histories/ intermodal_consumerpackagegoodsco.pdf.

[CombineNet, Inc., 2003c] CombineNet, Inc. CombineNet provides essential optimization to enhance sourcing for global food manufacturer, 2003. Case study white paper at www.CombineNet.com/CUSTOMERS/case_histories/ seafreight_consumerpackagegoodsco.pdf.

[CombineNet, Inc., 2003d] CombineNet, Inc. Leading auto manufacturer enhances sea freight allocations with CombineNet, 2003. Case study white paper at www.CombineNet.com/CUSTOMERS/case_histories/ seafreight_big3automotiveco.pdf.

[CombineNet, Inc., 2003e] CombineNet, Inc. Major industrial manufacturer identifies best strategy for optimizing sourcing with CombineNet, 2003. Case study white paper at www.CombineNet.com/CUSTOMERS/case_histories/ directmaterials_industrialmfgco.pdf).

[Conen and Sandholm, 2001] Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions: Extended abstract. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 256–259, Tampa, FL, October 2001. A more detailed description of the algorithmic aspects appeared in the IJCAI-2001 Workshop on Economic Agents, Models, and Mechanisms, pp. 71–80.

[Conen and Sandholm, 2002a] Wolfram Conen and Tuomas Sandholm. Differential-revelation VCG mechanisms for combinatorial auctions. In *AAMAS-02 workshop on Agent-Mediated Electronic Commerce (AMEC)*, Bologna, Italy, 2002.

[Conen and Sandholm, 2002b] Wolfram Conen and Tuomas Sandholm. Partial-revelation VCG mechanism for combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 367–372, Edmonton, Canada, 2002.

[Conitzer and Sandholm, 2002a] Vincent Conitzer and Tuomas Sandholm. Automated mechanism design: Complexity results stemming from the single-agent setting, 2002. Draft.

[Conitzer and Sandholm, 2002b] Vincent Conitzer and Tuomas Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 314–319, Edmonton, Canada, 2002.

[Conitzer and Sandholm, 2002c] Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 103–110, Edmonton, Canada, 2002.

[Conitzer and Sandholm, 2002d] Vincent Conitzer and Tuomas Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 392–397, Edmonton, Canada, 2002.

[Conitzer and Sandholm, 2003a] Vincent Conitzer and Tuomas Sandholm. Applications of automated mechanism design, 2003.

[Conitzer and Sandholm, 2003b] Vincent Conitzer and Tuomas Sandholm. Automated mechanism design for a self-interested designer. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, San Diego, CA, 2003. Poster paper.

[Conitzer and Sandholm, 2003c] Vincent Conitzer and Tuomas Sandholm. Automated mechanism design with a structured outcome space, 2003.

[Conitzer and Sandholm, 2003d] Vincent Conitzer and Tuomas Sandholm. Computational criticisms of the revelation principle, 2003. Draft.

[Conitzer and Sandholm, 2003e] Vincent Conitzer and Tuomas Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003.

[Conitzer *et al.*, 2003] Vincent Conitzer, Jerome Lang, and Tuomas Sandholm. How many candidates are needed to make elections hard to manipulate? In *Theoretical Aspects of Rationality and Knowledge (TARK IX)*, Bloomington, Indiana, USA, 2003.

[Conry *et al.*, 1991] Susan E Conry, K Kuwabara, Victor R Lesser, and R A Meyer. Multistage negotiation for distributed satisfaction. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1462–1477, 1991.

[d'Aspremont and Gérard-Varet, 1979] C d'Aspremont and L A Gérard-Varet. Incentives and incomplete information. *Journal of Public Economics*, 11:25–45, 1979.

[DeMartini *et al.*, 1999] C DeMartini, A Kwasnica, J Ledyard, and D Porter. A new and improved design for multi-object iterative auctions. Technical Report 1054, California Institute of Technology, Social Science, September 1999.

[Ephrati and Rosenschein, 1991] Eithan Ephrati and Jeffrey S Rosenschein. The Clarke tax as a consensus mechanism among automated agents. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 173–178, Anaheim, CA, 1991.

[Ephrati and Rosenschein, 1993] Eithan Ephrati and Jeffrey S Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 423–429, Chambery, France, 1993.

[Ephrati, 1994] Eithan Ephrati. A non-manipulable meeting scheduling system. In *Proc. 13th International Distributed Artificial Intelligence Workshop*, Lake Quinalt, Washington, July 1994. AAAI Press Technical Report WS-94-02.

[Fujishima *et al.*, 1999] Yuzo Fujishima, Kevin Leyton-Brown, and Yoav Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 548–553, Stockholm, Sweden, August 1999.

[Gibbard, 1973] A Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[Gonen and Lehmann, 2000] Rica Gonen and Daniel Lehmann. Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 13–20, Minneapolis, MN, October 2000.

[Green and Laffont, 1979] J Green and J-J Laffont. *Incentives in Public Decision Making*. Amsterdam: North-Holland, 1979.

[Groves, 1973] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[Hoos and Boutilier, 2001] Holger Hoos and Craig Boutilier. Bidding languages for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1211–1217, Seattle, WA, 2001.

[Hudson and Sandholm, 2002] Benoit Hudson and Tuomas Sandholm. Effectiveness of preference elicitation in combinatorial auctions. In *AAMAS-02 workshop on Agent-Mediated Electronic Commerce (AMEC)*, Bologna, Italy, 2002. Extended version: Carnegie Mellon University, Computer Science Department, CMU-CS-02-124, March. Also: Stanford Institute for Theoretical Economics workshop (SITE-02).

[Hudson and Sandholm, 2003a] Benoit Hudson and Tuomas Sandholm. Generalizing preference elicitation in combinatorial auctions. In *International Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 2003. (Poster presentation.).

[Hudson and Sandholm, 2003b] Benoit Hudson and Tuomas Sandholm. Using value queries in combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, San Diego, CA, 2003. (Poster presentation.).

[Jakobsson, 1995] M Jakobsson. Ripping coins for fair exchange. In *EUROCRYPT*, pages 220–230, Seattle, WA, 1995.

[Kalagnanam *et al.*, 2001] Jayant Kalagnanam, Andrew Davenport, and Ho Lee. Computational aspects of clearing continuous call double auctions with assignment constraints and indivisible demand. *Electronic Commerce Research Journal*, 1(3), 2001.

[Kothari *et al.*, 2003] Anshul Kothari, Tuomas Sandholm, and Subhash Suri. Solving combinatorial exchanges: Optimality via few partial bids. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, San Diego, CA, 2003. Poster presentation. Early version was accepted to the AAAI-02 workshop on Artificial Intelligence for Business.

[Kuhn, 1955] Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[Larson and Sandholm, 2001a] Kate Larson and Tuomas Sandholm. Bargaining with limited computation: Deliberation equilibrium. *Artificial Intelligence*, 132(2):183–217, 2001. Short early version appeared in the Proceedings of the National Conference on Artificial Intelligence (AAAI), pp. 48–55, Austin, TX, 2000.

[Larson and Sandholm, 2001b] Kate Larson and Tuomas Sandholm. Computationally limited agents in auctions. In *AGENTS-01 Workshop of Agents for B2B*, pages 27–34, Montreal, Canada, May 2001.

[Larson and Sandholm, 2001c] Kate Larson and Tuomas Sandholm. Costly valuation computation in auctions. In *Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 169–182, Sienna, Italy, July 2001.

[Larson and Sandholm, 2002] Kate Larson and Tuomas Sandholm. An alternating offers bargaining model for computationally limited agents. In *International Conference on Autonomous Agents and Multi-Agent Systems*, Bologna, Italy, July 2002.

[Larson and Sandholm, 2003] Kate Larson and Tuomas Sandholm. Miscomputing ratio: The social cost of selfish computing. In *International Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, 2003. Early version appeared in the AAAI-02 workshop on Game-Theoretic and Decision-Theoretic Agents, Edmonton, Canada.

[Lehmann *et al.*, 2002] Daniel Lehmann, Lidian Ita O'Callaghan, and Yoav Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):577–602, 2002.

[Leyton-Brown *et al.*, 2000a] Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 66–76, Minneapolis, MN, 2000.

[Leyton-Brown *et al.*, 2000b] Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. An algorithm for multi-unit combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Austin, TX, August 2000.

[Mas-Colell *et al.*, 1995] Andreu Mas-Colell, Michael Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[Matsubara and Yokoo, 2000] Shigeo Matsubara and Makoto Yokoo. Defection-free exchange mechanisms for information goods. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS)*, pages 183–190, Boston, MA, July 2000.

[Mu'alem and Nisan, 2002] Ahuva Mu'alem and Noam Nisan. Truthful approximate mechanisms for restricted combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 379–384, Edmonton, Canada, July 2002.

[Myerson and Satterthwaite, 1983] Roger Myerson and Mark Satterthwaite. Efficient mechanisms for bilateral exchange. *Journal of Economic Theory*, 28:265–281, 1983.

[Myerson, 1981] Roger B Myerson. Optimal auction design. *Mathematics of Operation Research*, 6:58–73, 1981.

[National Consumers League, 1999] National Consumers League. New NCL survey shows consumers are both excited and confused about shopping online, 1999. www.natlconsumersleague.org/ BeEWisepr.html, Oct. 20. Survey conducted by Opinion Research Corporation.

[Nisan and Ronen, 2000] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 242–252, Minneapolis, MN, 2000.

[Nisan and Ronen, 2001] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001. Early version in STOC-99.

[Nisan and Segal, 2003] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting Lindahl prices, 2003. Working Paper (version: March 2003).

[Nisan, 2000] Noam Nisan. Bidding and allocation in combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 1–12, Minneapolis, MN, 2000.

[Parkes and Ungar, 2000] David C Parkes and Lyle Ungar. Iterative combinatorial auctions: Theory and practice. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 74–81, Austin, TX, August 2000.

[Rassenti *et al.*, 1982] S J Rassenti, V L Smith, and R L Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell J. of Economics*, 13:402–417, 1982.

[Rothkopf *et al.*, 1998] Michael H Rothkopf, Aleksandar Pekeč, and Ronald M Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.

[Sandholm and Ferrandon, 2000] Tuomas Sandholm and Vincent Ferrandon. Safe exchange planner. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS)*, pages 255–262, Boston, MA, July 2000.

[Sandholm and Lesser, 1995a] Tuomas Sandholm and Victor R Lesser. Equilibrium analysis of the possibilities of unenforced exchange in multiagent systems. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 694–701, Montreal, Canada, August 1995.

[Sandholm and Lesser, 1995b] Tuomas Sandholm and Victor R Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS)*, pages 328–335, San Francisco, CA, June 1995. Reprinted in *Readings in Agents*, Huhns and Singh, eds., pp. 66–73, 1997.

[Sandholm and Lesser, 2001] Tuomas Sandholm and Victor R Lesser. Leveled commitment contracts and strategic breach. *Games and Economic Behavior*, 35:212–270, 2001. Special issue on AI and Economics. Early versions appeared as *Advantages of a Leveled Commitment Contracting Protocol* in the proceedings of the National Conference on Artificial Intelligence (AAAI), pp. 126–133, Portland, OR, 1996, and as a University of Massachusetts at Amherst, Computer Science Department technical report 95-72.

[Sandholm and Lesser, 2002] Tuomas Sandholm and Victor Lesser. Leveled commitment contracting: A backtracking instrument for multiagent systems. *AI Magazine*, 23(3):89–100, 2002.

[Sandholm and Suri, 2001a] Tuomas Sandholm and Subhash Suri. Market clearability. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1145–1151, Seattle, WA, 2001.

[Sandholm and Suri, 2001b] Tuomas Sandholm and Subhash Suri. Side constraints and non-price attributes in markets. In *IJCAI-2001 Workshop on Distributed Constraint Reasoning*, pages 55–61, Seattle, WA, 2001.

[Sandholm and Suri, 2002] Tuomas Sandholm and Subhash Suri. Optimal clearing of supply/demand curves. In *13th Annual International Symposium on Algorithms and Computation (ISAAC)*, Vancouver, Canada, November 2002. Also appeared in the proceedings of the AAAI-02 workshop on Agent-Based Technologies for B2B Electronic Commerce (AAAI Technical Report WS-02-01), pp. 15–22, Edmonton, Canada.

[Sandholm and Suri, 2003] Tuomas Sandholm and Subhash Suri. BOB: Improved winner determination in combinatorial auctions and generalizations. *Artificial Intelligence*, 145:33–58, 2003. Early version: Improved Algorithms for Optimal Winner Determination in Combinatorial Auctions and Generalizations. National Conference on Artificial Intelligence (AAAI-00), pp. 90–97, Austin, TX, July 31 – August 2.

[Sandholm and Wang, 2002] Tuomas Sandholm and Xiaofeng Wang. (Im)possibility of safe exchange mechanism design. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 338–344, Edmonton, Canada, 2002.

[Sandholm and Zhou, 2002] Tuomas Sandholm and Yunhong Zhou. Surplus equivalence of leveled commitment contracts. *Artificial Intelligence*, 142:239–264, 2002. Early versions appeared in ICMAS-00 and in the AAAI-99 Workshop on Negotiation: Settling conflicts and identifying opportunities.

[Sandholm *et al.*, 1999] Tuomas Sandholm, Sandeep Sikka, and Samphel Norden. Algorithms for optimizing leveled commitment contracts. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 535–540, Stockholm, Sweden, 1999. Extended version: Washington University, Department of Computer Science technical report WUCS-99-04.

[Sandholm *et al.*, 2001] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1102–1108, Seattle, WA, 2001.

[Sandholm *et al.*, 2002] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. Winner determination in combinatorial auction generalizations. In *International Conference on Autonomous Agents and Multi-Agent Systems*, pages 69–76, Bologna, Italy, July 2002. Early version appeared at the AGENTS-01 Workshop on Agent-Based Approaches to B2B, pp. 35–41, Montreal, Canada, May 2001.

[Sandholm, 1991] Tuomas Sandholm. A strategy for decreasing the total transportation costs among area-distributed transportation centers. In *Nordic Operations Analysis in Cooperation (NOAS): OR in Business*, Turku School of Economics, Finland, 1991.

[Sandholm, 1993] Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 256–262, Washington, D.C., July 1993.

[Sandholm, 1996] Tuomas Sandholm. *Negotiation among Self-Interested Computationally Limited Agents*. PhD thesis, University of Massachusetts, Amherst, 1996. Available at http:// www.cs.cmu.edu/ ˜sandholm/ dissertation.ps.

[Sandholm, 1997] Tuomas Sandholm. Unenforced E-commerce transactions. *IEEE Internet Computing*, 1(6):47–54, Nov–Dec 1997. Special issue on Electronic Commerce.

[Sandholm, 1998] Tuomas. Sandholm. Contract types for satisficing task allocation: I theoretical results. In *AAAI Spring Symposium Series: Satisficing Models*, pages 68–75, Stanford University, CA, March 1998.

[Sandholm, 2000a] Tuomas Sandholm. Agents in electronic commerce: Component technologies for automated negotiation and coalition formation. *Autonomous Agents and Multi–Agent Systems*, 3(1):73–96, 2000. Special Issue on Best of ICMAS 98.

[Sandholm, 2000b] Tuomas Sandholm. Approaches to winner determination in combinatorial auctions. *Decision Support Systems*, 28:165–176, 2000.

[Sandholm, 2000c] Tuomas Sandholm. Issues in computational Vickrey auctions. *International Journal of Electronic Commerce*, 4(3):107–129, 2000. Special Issue on Applying Intelligent Agents for Electronic Commerce. A short, early version appeared at the Second International Conference on Multi–Agent Systems (ICMAS), pages 299–306, 1996.

[Sandholm, 2002a] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135:1–54, January 2002. First appeared as an invited talk at the First International Conference on Information and Computation Economies, Charleston, SC, Oct. 25–28, 1998. Extended version appeared as Washington Univ., Dept. of Computer Science, tech report WUCS-99-01, January 28th, 1999. Conference version appeared at the International Joint Conference on Artificial Intelligence (IJCAI), pp. 542–547, Stockholm, Sweden, 1999.

[Sandholm, 2002b] Tuomas Sandholm. eMediator: A next generation electronic commerce server. *Computational Intelligence*, 18(4):656–676, 2002. Special issue on Agent Technology for Electronic Commerce. Early versions appeared in the Conference on Autonomous Agents (AGENTS-00), pp. 73–96, 2000; AAAI-99 Workshop on AI in Electronic Commerce, Orlando, FL, pp. 46–55, July 1999; and as a Washington University, St. Louis, Dept. of Computer Science technical report WU-CS-99-02, Jan. 1999.

[Sathi and Fox, 1989] A Sathi and Mark Fox. Constraint-directed negotiation of resource reallocations. In Michael N. Huhns and Les Gasser, editors, *Distributed Artificial Intelligence*, volume 2 of *Research Notes in Artificial Intelligence*, chapter 8, pages 163–193. Pitman, 1989.

[Satterthwaite, 1975] M A Satterthwaite. Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[Smith *et al.*, 2002] Trey Smith, Tuomas Sandholm, and Reid Simmons. Constructing and clearing combinatorial exchanges using preference elicitation. In *AAAI-02 workshop on Preferences in AI and CP: Symbolic Approaches*, pages 87–93, 2002.

[Smith, 1980] Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, December 1980.

[Sycara *et al.*, 1991] Katia Sycara, S Roth, Norman Sadeh, and Mark Fox. Distributed constrained heuristic search. *IEEE Transactions on Systems, Man, and Cybernetics*, Special issue on distributed artificial intelligence SMC-21(6):1446–1461, Nov/Dec 1991.

[Tennenholtz, 2000] Moshe Tennenholtz. Some tractable combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Austin, TX, August 2000.

[Vickrey, 1961] W Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[Vohra, 2001] Rakesh V. Vohra. Research problems in combinatorial auctions. Mimeo, version Oct. 29, 2001.

[Walsh and Wellman, 2000] William Walsh and Michael Wellman. Distributed quiescence detection in multiagent negotiation. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS)*, pages 317–324, Boston, MA, July 2000.

[Walsh *et al.*, 2000] William Walsh, Michael Wellman, and Fredrik Ygge. Combinatorial auctions for supply chain formation. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 260–269, Minneapolis, MN, 2000.

[Wurman and Wellman, 2000] Peter R Wurman and Michael P Wellman. AkBA: A progressive, anonymous-price combinatorial auction. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 21–29, Minneapolis, MN, October 2000.

[Yokoo, 2000] Makoto Yokoo. Algorithms for distributed constraint satisfaction: A review. *Autonomous Agents and Multi-Agent Systems*, 3(2):189–212, 2000.

[Zinkevich *et al.*, 2003] Martin Zinkevich, Avrim Blum, and Tuomas Sandholm. On polynomial-time preference elicitation with value queries. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, San Diego, CA, 2003.