Intellectual development statement

Vincent Conitzer

As required, this intellectual development statement is divided into three main sections: Research, Teaching, and Service. I am happy to provide any additional details on request; comments are also welcome. My CV can be found at http://www.cs.duke.edu/∼conitzer/cv.pdf.

1 Research

My research focuses on issues in the intersection of computer science (especially artificial intelligence and multiagent systems) and economics (especially microeconomic theory, game theory, social choice, and mechanism design). Each of these two disciplines can contribute significantly to the other. On the one hand, as computer systems become more interconnected, multiple parties must interact in the same environment and compete for scarce resources, which necessarily introduces economic phenomena. Here, economic theory can contribute to computer science—though the specific environments faced by computer scientists often require new contributions. On the other hand, in the past, deployed economic mechanisms (such as auctions and exchanges) have been designed to require very limited computing and communication resources, even though economic theory allows for much more powerful mechanisms in principle. Computer science can contribute to economics by allowing us to fully operationalize economic theory, resulting in more efficient as well as entirely novel mechanisms—though this requires the design of new algorithms as well as other contributions. I am devoted to exploring both of these directions.

Most of my research can be placed in at least one of the following four categories. These categories do not separate cleanly. More detail on these topics is given later in this statement.

1. *Making decisions based on the preferences of multiple agents.* This can be done using, for example, voting or auction mechanisms. I am especially interested in cases where the space of alternatives is combinatorial in nature, for example, when multiple items must be allocated or multiple issues require a decision. This leads to questions about how the agents' preferences should be represented and communicated, as well as nontrivial optimization problems for finding the best alternative. Application domains for such techniques include combinatorial auctions (which sell related items at the same time and are used in a.o. spectrum auctions and sourcing events), sponsored search auctions (which major search engines use to allocate advertising space), combinatorial voting, kidney exchanges, prediction markets, expressive markets for donating to charities, and many others.

2. *Algorithms and complexity results for game-theoretic solution concepts.* Game theorists study how an agent should act in the presence of other agents who have their own preferences. They have introduced a variety of solution concepts, including Nash equilibrium but also many others. I work on operationalizing these concepts, by designing algorithms for computing or learning the solution in specific games. Application domains for such techniques include playing games of imperfect information such as poker or Liar's Dice, but also the strategic allocation of resources in real-world security domains.

3. *(Automated) mechanism design for strategic agents.* In mechanisms such as the ones discussed under 1, a self-interested (game-theoretic) agent is sometimes better off reporting her preferences insincerely, in order to obtain an outcome that is better for her. However, we can design these mechanisms in

such a way that good outcomes result in spite of self-interested behavior. I am especially interested in automated mechanism design (where we identify good mechanisms by having a computer search through the space of possible mechanisms in an intelligent way), and the effects of the agents' computational limitations on the mechanism design process. We have used our automated mechanism design techniques to discover new results in economic theory.

4. *Mechanism design in highly anonymous environments.* Mechanism design has traditionally focused on the problem of agents misreporting their preferences. However, in highly anonymous environments such as the Internet, another often-used manipulation is for an agent to pretend to be multiple agents. This clearly has ramifications for online elections and rating systems, but also for other mechanisms such as auctions. I am working on extending the theory of mechanism design to address this type of manipulation.

Below, I summarize my research in each of these categories. In some cases, the decision to place a topic in one category rather than another was fairly arbitrary.

A quick note. Research statements by their nature tend to be self-centered and, unless the researcher is strictly devoted to a single topic (which I am not), dispersed over too large a variety of topics. If your interest is in learning more about some of the topics discussed here rather than in getting a general overview of my research, I recommend following the references to the original papers, especially the overview articles. While I have tried to cite very closely related work in this statement, this format does not allow me to give a truly thorough background.

1.1 Making decisions based on the preferences of multiple agents

(An article with the same title that I wrote for the Communications of the ACM [38] gives a more general and less self-centered introduction to this topic.) Much, but not all, of my work in this category concerns computational aspects of voting mechanisms, a topic on which I have worked since 2001. A voting mechanism allows a group of multiple agents to make joint decisions. In recent years, this topic has attracted a large number of researchers, resulting in the formation of the *computational social choice* research community, which first held its own separate workshop (COMSOC) in 2006 and has been flourishing since (in 2010, when I was program co-chair, COMSOC counted 92 participants). The computational social choice community (including me) studies questions such as the following.

1.1.1 How computationally hard is it to run voting rules?

In social choice theory, a voting mechanism (aka. voting rule) takes as input multiple rankings of a fixed set of alternatives. These rankings usually represent the preferences of multiple agents over the alternatives, but they can also represent, say, multiple search engines' rankings of a set of relevant pages. As output, the voting rule produces the winning alternative, or, more generally, an aggregate ranking of the alternatives. Many voting rules have been proposed, each characterized by different desirable properties. Some important rules are NP-hard to run—that is, given the input rankings, the winner or aggregate ranking is NP-hard to determine. In my opinion, the most important rule for which this is the case is the *Kemeny* rule, which has a useful interpretation as computing the maximum likelihood estimate of the "correct" ranking of the alternatives [156], as well as another nice axiomatization [157]. (The Kemeny rule was shown to be NP-hard to run in the 1980s by Bartholdi *et al.* [11]; its exact complexity—complete for parallel access to NP—was characterized by Hemaspaandra *et al.* [101].) We[1] showed how to use integer programming techniques for computing Kemeny rankings [39] which, to my knowledge, are still state of the art when used with the

[1] I will use "we" for any paper with multiple authors, and "I" for the occasional single-author paper. In the interest of space I will not mention all my very talented co-authors by name in the text; this does not mean that they do not deserve a large amount of the credit for the work. The students that I advise deserve particular credit; more detail on them can be found in the "mentoring" section.

right solver. (Incidentally, we have used this formulation in our department to get preliminary rankings of our Ph.D. applicants based on committee member rankings.) For a close relative of the Kemeny rule, the *Slater* rule, my techniques for computing the aggregate ranking [29] are also still state of the art (to my knowledge). In this paper, I also used these techniques to give a relatively simple direct proof of the NP-hardness of computing Slater rankings when there are no ties. (This is equivalent to the Minimum Feedback Arc Set problem on tournament graphs, which was conjectured to be NP-hard as early as 1992 [9]. The conjecture remained unproven until 2005, when a randomized reduction was given by Ailon *et al.* [1]. A later derandomization of this reduction by Alon finally proved the conjecture completely [2], briefly before my paper. There is also a fourth (2007) paper proving this result [21], which independently produced a similar proof to Alon's.)

1.1.2 How much communication is required to run voting rules?

To use a voting rule, the voters need to communicate enough information about their preferences (rankings) to determine the winner (or aggregate ranking). This is also known as the *preference elicitation* problem, where the voters are asked queries about their preferences until enough information is known to determine the winner. Voting rules that require less preference information to be communicated are preferred for a variety of reasons. We have characterized the communication complexity of the standard voting rules [58]. I have also studied how the number of queries can be decreased if preferences are known to lie in the class of *single-peaked preferences* [35] (this is related to later research that we performed on voting over intervals—*e.g.*, voting over the minimum and maximum sentence for a given offense [82]). We have also studied the effects of preference elicitation on strategic voting, and characterized the computational complexity of problems related to preference elicitation [46, 141]. Finally, we have characterized the *compilation complexity* of common voting rules [145]. (Compilation complexity, a notion related to communication complexity that was introduced in the context of voting by Chevaleyre *et al.* [25], concerns how many bits are required to summarize the votes of a subelectorate.)

1.1.3 How computationally hard is it to manipulate voting rules?

A voter is said to *manipulate* an election if she strategically misrepresents her preferences in order to obtain a better result for herself. Unfortunately, by a result known as the Gibbard-Satterthwaite theorem, every reasonable voting rule suffers from the problem that a voter can benefit from manipulating in some cases.[2] One line of research in the computational social choice community studies whether finding such opportunities for manipulation can be made computationally hard, so that even though they must exist, in practice voters will not be able to take advantage of them. Indeed, various voting rules have been shown to be NP-hard to manipulate under various circumstances, and we have contributed a number of these results [71, 51, 151, 150]. (The earliest results of this type date back to the 1980s, again by Bartholdi *et al.* [10]. They additionally considered the use of complexity as a barrier against undesirable behavior by the chair of the election [12], and we have also contributed such results in the context of combinatorial alternative spaces [43], a context that I will discuss shortly.)

However, this type of result is still unsatisfactory, because NP-hardness is a *worst-case* measure of hardness. Hence, even rules that are NP-hard to manipulate may be easy to manipulate *most of the time*. We have proved a number of results that indicate that this is indeed the case for existing rules, and moreover that this may be inevitable in general. For example, we have shown that if the combination of the voting rule and the distribution over instances satisfies some natural conditions, then a simple manipulation algorithm will find a manipulation most of the time [64]; that for an extremely general class of voting rules, the manipulators' ability to succeed is primarily determined by their number [142] (this result builds on related earlier work,

[2] We have also studied settings where the agents vote in sequence, and shown a general result that strategic behavior in this context can result in paradoxically bad outcomes, under every rule in a large class of voting rules [146]. (There is related work for the special case of the plurality rule by Desmedt and Elkind [78].)

for example by Procaccia and Rosenschein [125], but our results hold for a much more general class of rules, as illustrated by a later precise axiomatization that we obtained for this class [144]); that a version of the Gibbard-Satterthwaite impossibility result holds that characterizes how often purely random manipulations will succeed [143] (the first such result was by Friedgut, Kalai, and Nisan [85]); and finally, that there is an algorithm that effectively manipulates the class of positional scoring rules in an approximate sense (defined by Zuckerman *et al.* [160]) [150]. Taken together, these results cast significant doubt on the agenda of finding a voting rule that is hard to manipulate most of the time.

A good overview of the state of the art for these types of results is given by Faliszewski and Procaccia [81].

1.1.4 How should we vote when the space of alternatives is combinatorial in nature?

Often, a group of agents needs to make decisions on multiple interrelated issues. In this case, the preferences of an agent (voter) for one of the issues can depend on the decisions made on the other issues. For example, whether a voter wants one public resource (say, a tennis court) to be built can depend on whether another public resource (say, a swimming pool) is built. It is well known that voting over the issues separately can lead to terrible results [17, 109], but surprisingly little was known about how this should be addressed; we have been trying to address this in our work. Conceptually, we can consider every possible *joint* decision (consisting of a decision for every issue) to be an alternative, but ranking all these alternatives is not practical because there are exponentially many. Instead, we are now using languages from the AI literature for concisely representing preferences, such as CP-nets [14] (which are somewhat similar to Bayes nets). Using such representations in a voting framework opens up a variety of questions. How do we define a good voting rule for the representation? Can such a rule be executed in polynomial time? How much communication does this require? Can we make the rule robust to manipulation? I believe that such combinatorial alternative spaces represent the future of computational social choice, and that this is where the field may make its largest contributions to society: settings where issues are completely independent are exceedingly rare, but still we often decide on them as if they are. A computational perspective is key for going beyond the limited existing framework. We have already introduced some techniques for defining voting rules and computing their outcomes based on CP-nets [148, 149]. (The second paper follows the approach of determining the rule that produces the maximum likelihood estimate of the "correct" outcome—we have already studied this approach in detail in the non-combinatorial context [57, 44, 3]). We have also studied to what extent voting rules in combinatorial domains can be made *strategy-proof* (so that there is never an opportunity to manipulate) [147].

1.1.5 Other applications: auctions, donations, and externalities

While voting is a general methodology for reaching a decision based on the preferences of multiple agents, for specific applications, it is often preferable to use a special-purpose mechanism. For example, if the goal is to determine how to allocate a set of resources to the agents, a common approach is to auction off the resources. In a *combinatorial auction* (for an overview, see Cramton *et al.* [74]), the agents can directly bid on (express their preferences over) subsets of the resources. As in the context of voting, this leads to questions about how preferences should be represented (*bidding languages* [15, 115]) and communicated/elicited [120, 128], and how computationally hard it is to determine the outcome given the reported preferences (*winner determination* [110, 127]). (I recently wrote an article about the similarities between computer science research on combinatorial auctions and voting [37], as well as a book chapter on auctions [34].) These questions have received a significant amount of prior attention in the context of combinatorial auctions. Still, we have made some new contributions: we have introduced several graph-based bidding languages and studied the winner determination and preference elicitation problems for them [40, 131, 72]. (Essentially the same framework as in our paper [72] was simultaneously and independently proposed by Chevaleyre *et al.* [24], though with different results. An elegant generalization of the techniques in our paper [40] was recently given by Gottlob and Greco [89].) During graduate school, I also consulted for CombineNet, Inc., which gave me a nice perspective on real-world combinatorial (reverse) auctions.

As an example of a different special-purpose mechanism, we have introduced a methodology for letting donors make their donations to charities conditional on donations by other donors (who, in turn, can make their donations conditional) [70]. We have used this mechanism to collect money for Indian Ocean Tsunami and Hurricane Katrina victims. We have also introduced a more general framework for negotiation when one agent's actions have a direct effect (*externality*) on the other agents' utilities [69]. Both the charities and externalities methodologies require the solution of NP-hard optimization problems in general, but there are some natural tractable cases as well as effective MIP formulations. Recently, Ghosh and Mahdian [86] at Yahoo! Research extended our charities work, and based on this a web-based system for charitable donations was built at Yahoo!

1.2 Algorithms and complexity results for game-theoretic solution concepts

Computer scientists are increasingly confronted with settings where multiple self-interested agents interact in the same environment. In such settings, what is optimal for one agent to do in general depends on what the other agents do, leading to a tricky circularity. Game theory considers how agents should act under such conditions. It defines various solution concepts that correspond to various definitions of "rational" behavior. Algorithms for computing these solutions have various applications. They can be useful in artificial intelligence when an agent acts in a strategic domain with other agents/humans. For example, game theory has recently been very successful in computer (heads-up) poker, and I have used it to create an optimal program for a class of Liar's Dice games. More recently, such algorithms have started to be used in various real-world security and law enforcement applications. Since these applications are closely related to some of my research I will discuss them in more detail shortly.

Positive results in this area (efficient algorithms for finding solutions) are complemented by negative results: hardness results from complexity theory. These hardness results provide important guidance for the design of algorithms: for example, there is (presumably) little sense in trying to find a linear program for a problem that has been shown to be NP-hard.

1.2.1 Computing game-theoretic solutions in noncooperative normal-form and Bayesian games

Some of my work concerns the complexity of solving games according to standard concepts in noncooperative game theory (the more popular branch of game theory), specifically in normal-form games (the most basic representation of a game). We have characterized the computational complexity of a variety of problems related to finding Nash equilibria; in particular, we have shown that there is no efficient algorithm for finding approximately optimal equilibria (for various definitions of "optimal") unless P=NP [68]. This paper is standard reading in many algorithmic game theory courses; it significantly improves on earlier results by Gilboa and Zemel from the 1980s [88].[3] In addition, we have studied mixed integer programming techniques [130] as well as powerful preprocessing techniques [65] for computing Nash equilibria. We have also characterized the computational complexity of versions of (iterated) dominance [59], which correspond to even more basic notions of rationality in game theory. This paper significantly improves on earlier work by Gilboa, Kalai, and Zemel [87] in the early 1990s.

[3]These optimization problems are technically quite different from the problem of computing any single Nash equilibrium; a recent sequence of papers has resulted in the breakthrough result that computing a Nash equilibrium of a two-player game is PPAD-complete [75, 22, 23]. (I have shown that a classic algorithm in game theory, *fictitious play* [18], computes an approximate equilibrium; a result by Feder *et al.* [84] implies that this approximation guarantee approaches the best possible for a large class of algorithms—though other algorithms in this class are known that obtain the optimal guarantee exactly [76].) For three or more players, the problem of computing an exact Nash equilibrium (or an approximate one in a stronger sense) is FIXP-complete [80].

1.2.2 Commitment and security

In 2006, we published a paper that gave efficient algorithms as well as hardness results for computing *Stackelberg* strategies in various basic settings [62]. (One of these algorithms was also discovered by von Stengel and Zamir [136].) Stackelberg strategies are optimal when one of the players can commit to a strategy before the other player can make a decision. While our paper was theoretical in nature, it has been credited by Milind Tambe (at the University of Southern California) as a launching point for his group's work on computing Stackelberg strategies in real-world deployed security applications, including the optimal placement of checkpoints and canine units at Los Angeles International airport [121, 123] (the Transportation Security Administration (TSA) is now testing and evaluating the system for potential national deployment, which would be at over 400 airports), the assignment of Federal Air Marshals to flights [106, 134], the determination of patrolling strategies for the US Coast Guard, and potentially many other applications. For instances as large as the Federal Air Marshals application, it is difficult to scale general-purpose algorithms for computing Stackelberg strategies, so special-purpose algorithms are required. We have taken the general model of security games introduced by the USC group [106] and characterized the complexity of computing optimal Stackelberg strategies in variants of these games [107]. Also, together with the USC group, we have studied game-theoretic properties of these security games, including the relationship between Stackelberg and Nash strategies and the interchangeability of Nash equilibria [152]. In other work, we have proposed and evaluated algorithms for solving a game in which a seeker with flexible cameras searches for a fugitive hider that tries to avoid detection [98], as well as (together with the USC group) for solving a game where a defender places defensive resources on a graph to prevent an attacker from reaching targets of varying values (corresponding to the problem the Mumbai police face after the 2008 terrorist attacks) [104]. We are continuing our work on this type of topic as well as our collaboration with the USC group.

In addition, we are continuing our work on computing Stackelberg strategies in general games (not necessarily security games). We have characterized to what extent Stackelberg strategies can be approximated in polynomial time in Bayesian games [113]. We have also studied the complexity of computing Stackelberg strategies in extensive-form games [111]. Finally, we have designed and evaluated an iterative algorithm for computing equilibrium strategies in a model where there is uncertainty about whether the follower can observe the leader's mixed strategy before acting [108]; the extreme cases of this model correspond to the basic Stackelberg and Nash models.

1.2.3 Learning in games

If a game is played more than once, it is possible to learn something about the game or the opponent. For example, the payoff structure of the game may need to be learned; alternatively, we can learn to exploit the weaknesses of a suboptimal opponent. The topic of learning in games is complex and the work done in it is heterogeneous, but some of it is closely related to reinforcement learning in artificial intelligence. We have given the first learning algorithm that converges to a best response against a stationary opponent, but converges to Nash equilibrium in self-play, in arbitrary games [66]. (Before this algorithm, the algorithm that came closest—by Bowling and Veloso [16], in turn an improvement over an algorithm by Singh, Kearns, and Mansour [133]—only works for two-player, two-action games, and additionally needs to assume that the mixed strategies are directly observable.) We have also introduced a framework for learning to play zero-sum games with a bounded total loss (relative to optimal play) [49]. Additionally, we have shown how communication complexity can be used to provide lower bounds on the required time for learning [53] (a nice follow-up paper was later written by Hart and Mansour [100]). Finally, we have studied how to learn Stackelberg strategies when the follower's utility function is (initially) unknown [113].

1.2.4 Defining new game-theoretic solution concepts (and computing their solutions)

In the course of working on computing game-theoretic solutions, we have also introduced some new solution concepts. One of these is a parameterized solution concept that spans a spectrum between the two traditional solution concepts of dominance and Nash equilibrium (which correspond to the two extreme settings of the parameter) [60]. The parameter reflects the computational complexity of the concept: when its value is close to the extreme corresponding to dominance, the concept is computationally easy, but the problem becomes harder as the parameter increases towards the extreme corresponding to Nash equilibrium. At a minimum, this concept can be used as a preprocessing step for computing Nash equilibria, but I think it is interesting in it own right.

More recently, we have been working on a solution concept corresponding to "ethical" behavior [112]. The motivation here is that the self-interested behavior prescribed by game theory can, at times, be argued to violate basic ethical principles, such as taking advantage of trusting players. We defined a concept for two-player perfect-information games that is rooted in standard game-theoretic thinking but seems to correspond to a more ethical type of behavior. We also gave an efficient algorithm for computing solutions according to this concept. Generalizing the concept to wider classes of games is a challenging direction for future research.

1.2.5 Computing game-theoretic solutions in cooperative games

The other branch of game theory is *cooperative game theory*, in which coalitions of players can form binding contracts. While there is a continuing effort to unify noncooperative and cooperative game theory (the "Nash programme"), for now, cooperative game theory has its own separate solution concepts. We have introduced concise representation schemes for cooperative games, and given algorithms and proved complexity results for standard solution concepts, including the core and the Shapley value [61, 55]. (The representation in our paper [55] was later nicely generalized by Ieong and Shoham [102].) Cooperative game theory also involves some natural optimization problems, including the *coalition structure generation* problem—how to partition the agents into coalitions (teams) to maximize total utility. We have studied how to find the optimal coalition structure under several natural representations of the problem [116] (in contrast to existing work that treats the relevant function as a black box).

1.3 (Automated) mechanism design for strategic agents

In the context of voting, I already briefly mentioned the problem of an agent misreporting her preferences in order to obtain a better result for herself. The same problem occurs in other types of mechanism—for example, underbidding or overbidding in auctions. Other agents will anticipate this and change their own actions accordingly, *etc.*, resulting in very strategic behavior. In effect, the choice of mechanism defines a game, which we can then analyze using game-theoretic methods. However, we get to design the mechanism (*e.g.*, the voting or auction rules), and thereby the game. Ideally, we design it in such a way that desirable outcomes emerge in spite of the agents' strategic behavior. The standard approach, justified by a result known as the revelation principle, is to make it optimal for every agent to reveal her true preferences. A mechanism with this property is variously called *incentive compatible*, *truthful*, or *strategy-proof*.

1.3.1 Automated mechanism design

The traditional approach to mechanism design is to consider a general mechanism design problem—say, maximizing the expected revenue of an auction—and to obtain a mathematical characterization of the optimal mechanism that is as general as possible. When successful, this approach results in beautiful and powerful mechanisms. Unfortunately, some important problems resist attempts at a clean general solution—for example, the problem of designing revenue-maximizing combinatorial auctions. Moreover, in practice, often a mechanism is required for a particular idiosyncratic instance with a particular idiosyncratic objective. It does not seem that creating a whole general theory for each such instance is worthwhile.

With these motivations in mind, we have been advocating a somewhat different approach to mechanism design, *automated mechanism design*, where an optimal mechanism is designed specifically for a particular instance, by having a computer search through the space of possible mechanisms in an intelligent manner. We have proposed and implemented algorithms for a number of basic variants of the automated mechanism design problem, and supplemented these with computational complexity results [45, 48, 47, 56, 52, 129, 27, 28].[4] Automated mechanism design has featured very prominently in the Ph.D. research of at least three recently graduated students outside my group: Nathanael Hyafil (Toronto), Radu Jurca (EPFL; received the Victor Lesser Dissertation Award) [105], and Eugene Vorobeychik (Michigan; runner-up for the Victor Lesser Dissertation Award) [137].

Automated mechanism design should not be seen as an approach that competes with traditional mechanism design, but rather as a complementary way of thinking. Partial characterization results from traditional mechanism design can be very helpful in automated mechanism design. Conversely, one can put automated mechanism design in the service of traditional mechanism design, by using it to solve many specific instances so that a general solution can be conjectured and then proven. This is a way in which techniques from computer science can contribute to the discovery of new results in economic theory. I next discuss several successful examples of this (see also our AAAI NECTAR paper on this [93]).

1.3.2 Redistribution mechanisms

In many domains, we need to allocate one or more resources (aka. objects, items) among multiple agents. An auction is a natural mechanism for doing so. If there is a seller, then this seller receives the revenue of the auction. However, in many settings, there is no seller: for example, the objects may be a shared resource. For instance, the agents may be trying to allocate the use of their shared computing resources (over a given period of time) among themselves. If we use an auction in such a setting, it makes sense to redistribute any revenue that results from the auction among the agents. Unfortunately, doing so can significantly affect the incentives that the agents face in the auction: if the auction is strategy-proof without redistribution, it may lose its strategy-proofness when redistribution is added. In fact, it turns out that it is impossible to redistribute all the revenue of the well-known generalized Vickrey (VCG) auction without breaking strategy-proofness: to maintain strategy-proofness, some money needs to leave the system of the agents—that is, it needs to be thrown away! Still, we would like to throw away as little money as possible. Hence, the goal becomes to design a strategy-proof redistribution scheme that maximizes the amount of revenue redistributed.

We have characterized a redistribution scheme that maximizes the fraction of revenue redistributed in the worst case while maintaining strategy-proofness [92]. Perhaps surprisingly, for sufficiently large auctions this scheme also has better performance on average than a previous scheme (versions of which had been proposed on at least three different occasions [8, 124, 20]). The amount of money thrown away drops off exponentially fast in the number of agents. (Hervé Moulin independently discovered the same mechanism in a slightly less general setting as the solution to a slightly different problem [114].) We have also analyzed how to maximize expected redistribution with respect to a prior (while maintaining strategy-proofness in the strongest sense) [95]. Additionally, we have defined a notion of when one redstribution scheme dominates another, and given general approaches for finding a scheme that dominates a given scheme, which experimentally leads to large improvements in redistribution [91]. We have also shown that, in a slightly restricted setting, the undominated schemes coincide exactly with the schemes that are optimal in expectation for some distribution [4]. Finally, we have analyzed how by allocating the objects inefficiently (*i.e.*, in a way that does not maximize the value of the allocation), it is sometimes possible to redistribute much more of the revenue,

[4] Some work on automatically choosing the mechanism to use that preceded our work was done by [26], [19], and [122]. These works focused on setting a parameter of the mechanism (rather than searching through the space of all possible mechanisms), and evaluated the resulting mechanism based on agents that they evolved with the mechanism (rather than requiring truthfulness).

so that the net effect is a significant increase in welfare [90], though many open questions remain here. (See also follow-up work by de Clippel *et al.* [77].)

What is (to me) perhaps even more interesting than the redistribution mechanisms themselves is the way in which we obtained them: our approach (described more precisely in the papers) has generally been to solve small instances by computer using search/optimization techniques, then discover the pattern in the solution and prove that this pattern gives the general solution. This is, to my knowledge, the most successful example so far of using automated mechanism design to prove general theorems in mechanism design, which has caught the attention of some economists.

1.3.3 Mechanism design without money

While the redistribution mechanisms described above try to minimize the amount of money that must leave the system of the agents, they nevertheless rely quite heavily on the *ability* of the agents to make payments, including to each other. However, in some domains, the agents do not have the ability to make payments at all. For example, in some domains this is illegal (*e.g.*, kidney exchanges); in others, it may simply be the case that no currency has been established yet (*e.g.*, peer-to-peer networks). What can be done in such domains?

In many of our early papers on the basic variants of automated mechanism design (listed above), we studied how hard it is to find optimal mechanisms in settings without payments. More recently, we have also studied the question of finding approximately optimal mechanisms that do not use money in more specific contexts where items need to be allocated. (For a nice general presentation of the agenda of designing approximately optimal mechanisms that do not use money, see Procaccia and Rosenschein [126].) We have studied this in the context where the same item needs to be allocated to the same agents in every period (*e.g.*, who gets to hold the remote / use the supercomputer today), and shown that in this context, in spite of agents' strategic behavior, it is possible to come quite close to efficient allocation: the intuition is that agents can effectively "pay" to receive the item today by giving up the item in the future [97]. We have also studied a single-period model where multiple items must be allocated: here, too, we can sometimes come reasonably close to efficiency—the intuition is that agents can pay for one item by giving up their claim to another item [96]. These two papers are additional examples where the automated mechanism design approach has been successful. We have also proposed a generalization of the Vickrey auction that can base the outcome on any characteristics of the bids, not necessarily their monetary amounts [99].

1.3.4 Mechanism design for bounded agents

The theory of mechanism design assumes that agents behave strategically optimally—that is, according to solution concepts defined in game theory. This assumption facilitates theoretical work: specifically, it allows one to prove the *revelation principle*, which states that only considering incentive compatible mechanisms is without loss of generality. This is arguably the fundamental building block of the theory of mechanism design. However, the assumption of game-theoretically perfect behavior on the agents' part is not always reasonable, for example because it would in some cases require the agents to solve computationally hard problems. We have already discussed the computational hardness of strategic voting, as well as of computing solutions of games in general, above. If agents are not able to behave in a game-theoretically optimal way, the revelation principle collapses. Specifically, we have shown that in some settings, there is a mechanism that is not incentive compatible that will perform as well as the optimal incentive compatible mechanism if agents behave in a game-theoretically optimal way, and strictly better if they do not [54]. This opens up the possibility of a generalized theory of mechanism design that can deal with computational limitations on the part of the agents. There has recently been some nice follow-up work on this [119].

These observations also led us to propose a new approach to designing mechanisms that we called *incremental mechanism design* [67]. Here, the idea is to start with a naïve mechanism that would work well if agents did not misreport their preferences. Then, we incrementally modify the mechanism to make it "more" incentive compatible, removing some of the opportunities for manipulation. This corresponds to the real-world process in which mechanisms are often patched to address particularly obvious opportunities for manipulation (but not necessarily other kinds of manipulation that are difficult to discover by the agents). We have shown that this approach can be used to rederive well-known incentive compatible mechanisms, but also well-known mechanisms that are not completely incentive compatible but that do address particularly obvious kinds of manipulation. For example, the methodology can be used to transform a naïve voting rule into one that uses a runoff round between the final two alternatives.

1.4 Mechanism design in highly anonymous environments

The current theory of mechanism design is primarily concerned with the possibility that agents lie about their preferences; a mechanism is strategy-proof if lying is never beneficial to the agent. However, in open, anonymous environments such as the Internet, other types of manipulation are possible. Specifically, a single agent can participate in the mechanism multiple times under multiple identifiers (*e.g.*, e-mail addresses). (Compare the notion of a Sybil attack [79].) One way to address this is to design *false-name-proof* mechanisms, a concept proposed by Yokoo *et al.* [154, 155], under which using more than one identifier is never beneficial to the agent. Many of the standard strategy-proof mechanisms are not false-name-proof. (There is some flexibility in the precise definition of false-name-proofness: we have pointed out that the standard definition may still leave room for manipulation if the manipulator can abandon identifiers, and a stronger definition can be given that rules this out [94].)

I recently gave a complete characterization of false-name-proof voting rules (that also satisfy some other minimal conditions, namely, they treat all alternatives symmetrically and it never hurts a voter to participate) [31]. Unfortunately, this is mostly a negative result, because the rules satisfying these properties end up choosing the winner almost completely at random. However, this result assumes that casting additional votes is completely costless. In another paper [138], we show that if we, more realistically, assume that casting an additional vote comes at a small cost (for example, representing the voter's effort in opening a new e-mail account), then the optimal false-name-proof voting rule changes into one that has much more desirable properties (such as choosing the majority winner in the limit as more votes are added).

In many settings, there are no good false-name-proof mechanisms. Besides the result for voting mentioned above, we have recently proven a tight bound on the worst-case efficiency of false-name-proof combinatorial auctions, and the corresponding efficiency is low enough that this must be viewed as a negative result [103]. Hence, it becomes important to investigate alternative models that may lead to more positive results (such as the model of costly identifiers mentioned above). An extreme solution is to give up on anonymity altogether, by verifying everyone's real-world identity. I have shown [30] that it is not necessary to go quite that far: based on the outcome of the mechanism (*e.g.*, auction or election), it is possible to selectively verify the identities of only some of the participating agents, in such a way that no agent has an incentive to use multiple identifiers. The paper gives a precise characterization of how much verification needs to be done, and algorithms for deciding which agents to verify. Another direction that we have pursued is to make use of social network structure [42]. The idea is that, while it is easy to create false nodes in a social network, it is not easy to get other nodes to link to these false nodes, and so we have some chance of detecting / preventing this type of manipulation. This high-level idea has been pursued before in the systems literature, notably by Yu *et al.* [159, 158], in the related context of preventing Sybil attacks. We show how to leverage this idea in a strict mechanism design context, as well as how to integrate it with the idea of limited verification discussed above.

I have also tried to create a methodology for preventing multiple account sign-ups by a single person altogether, by creating a way to randomly generate a memory test, which the user needs to pass to obtain an account, in such a way that it is easy to pass one of these tests, but difficult to pass a second one because the user becomes confused with the first test that she took [33]. Unfortunately, as the experimental results on human subjects in that paper show, those tests do not yet work well enough to be practical.

Anonymity issues also come up in cooperative game theory, where multiple agents working in a team each bring some resources to the table; this results in some value (profit) that is generated by the team and needs to be distributed among the agents. The standard game-theoretic schemes for distributing the value are vulnerable to false-name manipulations: for example, an agent with two resources can sometimes pretend to be two agents with one resource each, in a way that results in a higher total value distribution to these two fictitious agents. With Makoto Yokoo and his group (who pioneered the study of false-name-proofness), we have proposed various schemes for distributing the value that are not vulnerable to such manipulations, that is, they are anonymity-proof [153, 118, 117]. More recent work on false-name manipulation in weighted voting games by Bachrach and Elkind [7] and Aziz and Paterson [6] also fits in this general framework.

For more background on false-name-proofness, please see our general overview article [73].

1.5 Various other research

I have recently become interested in *prediction markets* [140]. A prediction market is a market organized around securities that pay off if a particular event happens; usually, the purpose of the prediction market is to assess the probability that the event happens, based on the market prices of the securities. (For example, something can be inferred about the probability of an event from betting odds.) We have created a hybrid of a prediction market and a Turing test [135], where the users can buy securities (place bets) on whether they are talking to a human or a bot, resulting in a quantitative assessment of how human-like a particular bot is [83]. (The game can be played online at http://turingtrade.org/.) I have also introduced a model of prediction markets, based on *proper scoring rules*, that fits more naturally into the framework of mechanism design [36]. This paper also introduces natural prediction market mechanisms in this framework, draws connections to cooperative game theory, and discusses how practical prediction markets can be designed based on these mechanisms. Finally, we have studied the problem that prediction markets can in principle give incentives to participants to perform undesirable actions in the real world—for example, a market attempting to predict terrorist attacks can in principle incentivize terrorism—and analyzed market mechanisms that do not introduce such incentives [132].

While a popular heuristic in games with chance is to maximize expected winnings, in reality sometimes the goal is just to have greater winnings than one's opponent(s) in the end. We recently completed a game-theoretic study of how to act optimally in such environments [139].

The VCG mechanism is a standard, general-purpose mechanism. It is well-known to be vulnerable to collusion by multiple agents, and have bizarre revenue properties [5]. We investigated the extent of these problems and their complexity in combinatorial auctions and their variants [63].

Learning is important not only in game theory, but also in mechanism design. For example, if the distribution of agents' preferences is unknown, the designer can try to learn this distribution (or at least enough to determine the optimal mechanism) by repeatedly changing the mechanism. We experimentally compared various learning approaches in settings with a single agent per period [41].

We have also shown that an auction known as the *adaptive clinching auction* satisfies the property that a bidder cannot gain from underreporting her budget, which allows us to make it incentive compatible with respect to the budget as well as to derive other properties [13].

As an example of work that does not involve game theory, mechanism design, social choice, multiagent systems, and the like, we characterized the computational complexity of various basic metareasoning problems (reasoning about how to reason optimally) [50]. I recently presented a discussion of this work and its implications [32].

2 Teaching

I consider teaching to be an essential and highly rewarding part of my career. My primary teaching mission is to teach courses in the intersection of computer science and (micro)economics that prepare students to do research in this interdisciplinary area, as well as to apply these methods in industry. I am also happy to teach other courses, such as artificial intelligence or linear/integer programming, as the needs of the department require.

As for teaching philosophy, I try to make lectures as interactive as possible by encouraging questions and frequently testing students' understanding by asking them to solve small example problems. This is partially driven by selfish motives, as I am more comfortable teaching with some feedback. However, I also believe that this approach helps the students stay engaged. In any case, without such interaction one could argue that I would not be adding much value relative to a video posted online (or a well-written book).

I make all my course materials, including slides and lecture notes, publicly available on the Web. Based on anecdotal evidence, it appears that students, researchers, and faculty both inside and outside of Duke have benefited from this. Below I describe some of my courses (I have taught versions of some of these courses under different names earlier).

In my graduate course *Computational Microeconomics: Game Theory, Social Choice, and Mechanism Design*, I first introduce students to key concepts in microeconomic theory, including basic game theory, social choice, mechanism design, and auctions. Along the way, we study computational aspects of these concepts. The second part of the course focuses on individual topics in this interdisciplinary area. A student who has successfully completed this course should have a good basic understanding of the area and be ready to delve into a specific topic to start doing research. (The last time I taught this course it placed me among the top 5% of instructors at Duke by evaluations.) I have also previously taught a Topics course that goes into more detail on some selected individual topics.

My undergraduate course *CPS 173: Computational Microeconomics* covers similar topics, but from a somewhat more applied perspective. A student who has successfully completed this course should have a basic background in the design of new, computationally oriented marketplaces, and should have acquired some tools to put these ideas into practice. A challenging aspect of this course is that I require no previous programming background; instead, the programming assignments only require the use of a standard linear/integer programming package and its modeling language. This allows students without a computer science background (for instance, economics students) to take the course and be exposed to basic ideas from computer science. While the computer science students undoubtedly have an advantage on the programming assignments, the advantage is not that great because most of the computer science students tend to be unfamiliar with linear programming. (Of course, the economics students have an advantage in other parts of the course.) I believe that this course represents an unusual but effective way of introducing certain students to computer science; I have given several invited talks on this, including at the 2008 AAAI Spring Symposium on Using AI to Motivate Greater Participation in Computer Science and a 2010 ARTSI faculty workshop.

Motivated by the increasing usage of linear and integer programming in various areas of computer science, combined with the observation that many computer scientists are not very familiar with these techniques, I have also been teaching a graduate course specifically on these techniques. I enjoy teaching this course; unlike my other courses, I teach it entirely on the board. I have been very impressed with the great variety of topics to which the students have applied these techniques in their projects.

I have also taught artificial intelligence, both at the graduate level (the qual course) and at the undergraduate level. In both cases, the course gave an introduction to a variety of artificial intelligence techniques, ranging from the classical to the modern. A student who has successfully completed these courses should have a basic understanding of AI and be ready to take more advanced courses, or to engage in research with significant guidance. I enjoy teaching these courses because it helps to keep me connected to the broader spectrum of techniques in AI, which remains a core interest of mine.

Taken over all my course evaluations during my first five years at Duke, the average "quality of course" score was 4.60/5.00, and the average "quality of instructor" score was 4.70/5.00.

3 Mentoring

I greatly enjoy working on research with students, and helping them develop into mature researchers. I am extremely fortunate to work (and have worked) with very talented graduate as well as undergraduate students. Two people have graduated from my group with a Ph.D. degree:

- **Mingyu Guo** received his Ph.D. in Computer Science in 2010 after working with me on automated mechanism design, redistribution mechanisms, mechanism design without payments, and false-name-proof mechanisms. His dissertation title was *Computationally Feasible Approaches to Automated Mechanism Design* and he received the department's Outstanding Ph.D. Dissertation Award for 2010 (as well as the department's nomination for the ACM Dissertation Award). He will start as a Lecturer (in the UK sense) in the University of Liverpool's Economics and Computation Group.

- **Liad (Leo) Wagman** (co-advised with Curtis Taylor in Economics) received his Ph.D. in Economics in 2009 after working with me on false-name-proofness when there is a cost to using additional identifiers, paying for anonymity in the context of price discrimination, and playing to end up with more money than one's opponent, His dissertation title was *Essays on Privacy, Information, and Anonymous Transactions*. He now holds a tenure-track position in the business school of the Illinois Institute of Technology.

I currently supervise three computer science Ph.D. students:

- **Dmytro (Dima) Korzhyk** is working with me on applying algorithms for computing game-theoretic solutions to security domains.

- **Joshua (Josh) Letchford** is working with me on computing, learning, and analyzing optimal mixed strategies to commit to, and has also worked with me on ethical solution concepts in game theory and using social networks to prevent false-name manipulation.

- **Lirong Xia** is working with me on a large number of topics in computational social choice and voting, particularly on voting in combinatorial domains, and the analysis and complexity of strategic manipulation by the voters.

I have also supervised two Master's students, **Peter Franklin** (title: *Internet-based Fault Recovery for Home Wireless LANs*, winner of the department's Outstanding Master's Award for 2010), and **Joseph Farfel** (title: *Getting agents to agree on numbers: Aggregating value ranges and Turing Trade*). I have also worked with a number of undergraduates, including **Peng Shi** (who worked with me on prediction markets and related topics; he was a CRA Outstanding Undergraduate Research Award Finalist and is starting his Ph.D. studies at MIT), **Matthew Rognlie** (who worked with me on interpreting voting rules as maximum likelihood estimators and repeated games in which players can restart the game; he was a Duke Faculty Scholar and is also starting his Ph.D. studies at MIT), **Maggie Bashford** (who worked with me on how computationally hard it typically is to control elections), and **Everett Wetchler** (who did a senior-year independent study on computer poker with me).

I do not believe in having a one-size-fits-all strategy for mentoring students, as each student comes in with particular strengths and weaknesses. My goal is to make myself as available as possible to them and help them where I can in preparing them for their future careers.

4 Service

This final section gives a brief overview of some of my other professional activities.

4.1 Secondary appointment in economics

I hold a secondary appointment in the economics department. There, I interact closely with the microeconomic theory faculty: my students and I regularly attend, and sometimes present in, their seminars. We also interact with faculty at Fuqua (Duke's business school) and attend their seminars. I coordinate interest in the intersection of computer science and economics at Duke, among other things by maintaining the mailing list `cs-econ@duke.edu` and sending announcements about relevant talks.

Our group also tries to maintain visibility in economics and operations research more generally. We have published two papers in Games and Economic Behavior (the top game theory journal), and attend conferences such as the World Congress of the Game Theory Society, the Stony Brook Game Theory Festival, and the INFORMS Annual Meeting. I am occasionally invited to give a talk in an economics department or conference (recently, at Caltech, Rice, and Guanajuato).

4.2 Service to the computer science department

So far, I have served on the computer science department's graduate admissions, faculty search, graduate program, and undergraduate curriculum committees. For the graduate admissions committee, Jun Yang and I put together some software for ranking the candidates based on committee member scores, which is still in use. Ron Parr, Carlo Tomasi, and I coordinate the DRIV (AI) seminar; Kamesh Munagala and I (and graduate student Sayan Bhattacharya) coordinate the Computational Microeconomics Reading Group. I have served on the committees of a number of students, and I served as the site judge at Duke for the ACM Collegiate Programming Contest, MidAtlantic region, in 2006.

4.3 Selected service to the research community

I spent several years as the editor-in-chief of SIGecom Exchanges, the newsletter for the ACM Special Interest Group on Electronic Commerce. (This special interest group consists primarily of computer scientists with a strong interest in economics.) My goal was to change the focus of the newsletter from full-length articles to shorter letters that give a brief summary of recent results. I believe that this has been successful: the most prominent researchers in the area are now publishing short letters (and the occasional full-length article) in the Exchanges. I also created a puzzle for each issue, and these puzzles were quite popular and the tradition is being continued. The author(s) of the nicest solution get(s) to publish it in the subsequent issue. We are now trying to start a new journal on Economics and Computation (currently under review by ACM) for which Preston McAfee and I are the proposed editors-in-chief.

Besides this, I also serve as an Associate Editor of the Journal of Artificial Intelligence Research and the Journal of Autonomous Agents and Multi Agent Systems, and on the editorial board of the Artificial Intelligence journal. I will be program co-chair of AAMAS 2012 and served as program co-chair of COMSOC 2010. I serve on the IFAAMAS board and on the steering committee of AMMA; I have served as an Area Chair for IJCAI and AAAI, on the Senior Program Committee of AAMAS, EC, AISTATS, and UAI, and in various other organizing / program committee roles. More details on my editorial and program committee service can be found on my CV.

I maintain the main mailing list for the computational social choice community, `comsoc@duke.edu`. We have given several tutorials at conferences, including on computational social choice, automated mechanism design, and computing game-theoretic solutions. I served on the Ph.D. committee of Troels Bjerre Sørensen (University of Aarhus), and as an M.S. thesis evaluator for Reshef Meir and Michael Zuckerman (Hebrew University).

References

[1] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 684–693, 2005.

[2] Noga Alon. Ranking tournaments. *SIAM Journal of Discrete Mathematics*, 20:137–142, 2006.

[3] Mehmet Serkan Apaydin, Vincent Conitzer, and Bruce Randall Donald. Structure-based protein NMR assignments using native structural ensembles. *Journal of Biomolecular NMR*, 40(4):263–276, April 2008.

[4] Krzysztof Apt, Vincent Conitzer, Mingyu Guo, and Evangelos Markakis. Welfare undominated Groves mechanisms. In *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, pages 426–437, Shanghai, China, 2008.

[5] Lawrence M. Ausubel and Paul Milgrom. The lovely but lonely Vickrey auction. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 1, pages 17–40. MIT Press, 2006.

[6] Haris Aziz and Mike Paterson. False name manipulations in weighted voting games: splitting, merging and annexation. In *Proceedings of the Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 409–416, Budapest, Hungary, 2009.

[7] Yoram Bachrach and Edith Elkind. Divide and conquer: False-name manipulations in weighted voting games. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 975–982, Estoril, Portugal, 2008.

[8] Martin J. Bailey. The demand revealing process: to distribute the surplus. *Public Choice*, 91:107–126, 1997.

[9] Jorgen Bang-Jensen and Carsten Thomassen. A polynomial algorithm for the 2-path problem for semicomplete digraphs. *SIAM Journal of Discrete Mathematics*, 5(3):366–376, 1992.

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

[11] John Bartholdi, III, Craig Tovey, and Michael Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.

[12] John Bartholdi, III, Craig Tovey, and Michael Trick. How hard is it to control an election? *Math. Comput. Modelling*, 16(8-9):27–40, 1992. Formal theories of politics, II.

[13] Sayan Bhattacharya, Vincent Conitzer, Kamesh Munagala, and Lirong Xia. Incentive compatible budget elicitation in multi-unit auctions. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 554–572, 2010.

[14] Craig Boutilier, Ronen Brafman, Carmel Domshlak, Holger Hoos, and David Poole. CP-nets: a tool for representing and reasoning with conditional ceteris paribus statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

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

[16] Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[17] S. Brams, D. Kilgour, and W. Zwicker. The paradox of multiple elections. *Social Choice and Welfare*, 15(2):211–236, 1998.

[18] George W. Brown. Iterative solutions of games by fictitious play. In T.C. Koopmans, editor, *Activity Analysis of Production and Allocation*. New York: Wiley, 1951.

[19] Andrew Byde. Applying evolutionary game theory to auction mechanism design. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 192–193, San Diego, CA, USA, 2003. Poster paper.

[20] Ruggiero Cavallo. Optimal decision-making with minimal waste: Strategyproof redistribution of VCG payments. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 882–889, Hakodate, Japan, 2006.

[21] Pierre Charbit, Stéphan Thomassé, and Anders Yeo. The minimum feedback arc set problem is NP-hard for tournaments. *Combinatorics, Probability and Computing*, 16(1):1–4, 2007.

[22] Xi Chen and Xiaotie Deng. Settling the complexity of two-player Nash equilibrium. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 261–272, 2006.

[23] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Computing Nash equilibria: Approximation and smoothed complexity. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 603–612, 2006.

[24] Yann Chevaleyre, Ulle Endriss, Sylvia Estivie, and Nicolas Maudet. Multiagent resource allocation with k-additive utility functions. In *Workshop on Computer Science and Decision Theory*, 2004.

[25] Yann Chevaleyre, Jérôme Lang, Nicolas Maudet, and Guillaume Ravilly-Abadie. Compiling the votes of a subelectorate. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 97–102, Pasadena, CA, USA, 2009.

[26] Dave Cliff. Evolution of market mechanism through a continuous space of auction-types. Technical Report HPL-2001-326, HP Labs, 2001.

[27] Vincent Conitzer. Computational aspects of mechanism design. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1642–1643, Pittsburgh, PA, USA, 2005. Doctoral Consortium Abstract.

[28] Vincent Conitzer. *Computational aspects of preference aggregation*. PhD thesis, Carnegie Mellon University, 2006.

[29] Vincent Conitzer. Computing Slater rankings using similarities among candidates. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 613–619, Boston, MA, USA, 2006. Early version appeared as IBM RC 23748, 2005.

[30] Vincent Conitzer. Limited verification of identities to induce false-name-proofness. In *Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 102–111, Brussels, Belgium, 2007.

[31] Vincent Conitzer. Anonymity-proof voting rules. In *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, pages 295–306, Shanghai, China, 2008.

[32] Vincent Conitzer. Metareasoning as a formal computational problem. In *The AAAI-08 Workshop on Metareasoning: Thinking about Thinking*, Chicago, IL, USA, 2008.

[33] Vincent Conitzer. Using a memory test to limit a user to one account. In *Agent-Mediated Electronic Commerce (AMEC) workshop*, Estoril, Portugal, 2008.

[34] Vincent Conitzer. Auction protocols. In Mikhail Atallah and Marina Blanton, editors, *Algorithms and Theory of Computation Handbook*. CRC Press, Taylor & Francis Group, 2009.

[35] Vincent Conitzer. Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, 35:161–191, 2009. Early version in AAMAS-07.

[36] Vincent Conitzer. Prediction markets, mechanism design, and cooperative game theory. In *Proceedings of the 25th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 101–108, Montreal, Canada, 2009.

[37] Vincent Conitzer. Comparing multiagent systems research in combinatorial auctions and voting. *Annals of Mathematics and Artificial Intelligence*, 2010. To appear.

[38] Vincent Conitzer. Making decisions based on the preferences of multiple agents. *Communications of the ACM*, 53(3):84–94, 2010.

[39] Vincent Conitzer, Andrew Davenport, and Jayant Kalagnanam. Improved bounds for computing Kemeny rankings. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 620–626, Boston, MA, USA, 2006.

[40] Vincent Conitzer, Jonathan Derryberry, and Tuomas Sandholm. Combinatorial auctions with structured item graphs. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 212–218, San Jose, CA, USA, 2004.

[41] Vincent Conitzer and Nikesh Garera. Learning algorithms for online principal-agent problems (and selling goods online). In *International Conference on Machine Learning (ICML)*, pages 209–216, Pittsburgh, PA, USA, 2006.

[42] Vincent Conitzer, Nicole Immorlica, Joshua Letchford, Kamesh Munagala, and Liad Wagman. False-name-proofness in social networks. In *Proceedings of the Sixth Workshop on Internet and Network Economics (WINE)*, Stanford, CA, USA, 2010.

[43] Vincent Conitzer, Jérôme Lang, and Lirong Xia. How hard is it to control sequential elections via the agenda? In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 103–108, Pasadena, CA, USA, 2009.

[44] Vincent Conitzer, Matthew Rognlie, and Lirong Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 109–115, Pasadena, CA, USA, 2009.

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

[46] 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, AB, Canada, 2002.

[47] Vincent Conitzer and Tuomas Sandholm. Applications of automated mechanism design. In *UAI-03 workshop on Bayesian Modeling Applications*, Acapulco, Mexico, 2003.

[48] Vincent Conitzer and Tuomas Sandholm. Automated mechanism design: Complexity results stemming from the single-agent setting. In *Proceedings of the 5th International Conference on Electronic Commerce (ICEC-03)*, pages 17–24, Pittsburgh, PA, USA, 2003.

[49] Vincent Conitzer and Tuomas Sandholm. BL-WoLF: A framework for loss-bounded learnability in zero-sum games. In *International Conference on Machine Learning (ICML)*, pages 91–98, Washington, DC, USA, 2003.

[50] Vincent Conitzer and Tuomas Sandholm. Definition and complexity of some basic metareasoning problems. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1099–1106, Acapulco, Mexico, 2003.

[51] 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)*, pages 781–788, Acapulco, Mexico, 2003.

[52] Vincent Conitzer and Tuomas Sandholm. An algorithm for automatically designing deterministic mechanisms without payments. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 128–135, New York, NY, USA, 2004.

[53] Vincent Conitzer and Tuomas Sandholm. Communication complexity as a lower bound for learning in games. In *International Conference on Machine Learning (ICML)*, pages 185–192, Banff, Alberta, Canada, 2004.

[54] Vincent Conitzer and Tuomas Sandholm. Computational criticisms of the revelation principle. In *The Conference on Logic and the Foundations of Game and Decision Theory (LOFT-04)*, Leipzig, Germany, 2004.

[55] Vincent Conitzer and Tuomas Sandholm. Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 219–225, San Jose, CA, USA, 2004.

[56] Vincent Conitzer and Tuomas Sandholm. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 132–141, New York, NY, USA, 2004.

[57] Vincent Conitzer and Tuomas Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 145–152, Edinburgh, UK, 2005.

[58] Vincent Conitzer and Tuomas Sandholm. Communication complexity of common voting rules. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 78–87, Vancouver, BC, Canada, 2005.

[59] Vincent Conitzer and Tuomas Sandholm. Complexity of (iterated) dominance. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 88–97, Vancouver, BC, Canada, 2005.

[60] Vincent Conitzer and Tuomas Sandholm. A generalized strategy eliminability criterion and computational methods for applying it. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 483–488, Pittsburgh, PA, USA, 2005.

[61] Vincent Conitzer and Tuomas Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170(6-7):607–619, 2006. Earlier version appeared in IJCAI-03, pages 613–618.

[62] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 82–90, Ann Arbor, MI, USA, 2006.

[63] Vincent Conitzer and Tuomas Sandholm. Failures of the VCG mechanism in combinatorial auctions and exchanges. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 521–528, Hakodate, Japan, 2006.

[64] Vincent Conitzer and Tuomas Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 627–634, Boston, MA, USA, 2006.

[65] Vincent Conitzer and Tuomas Sandholm. A technique for reducing normal-form games to compute a Nash equilibrium. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 537–544, Hakodate, Japan, 2006.

[66] Vincent Conitzer and Tuomas Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43, May 2007. Special Issue on Learning and Computational Game Theory. Early version in ICML-03.

[67] Vincent Conitzer and Tuomas Sandholm. Incremental mechanism design. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1251–1256, Hyderabad, India, 2007.

[68] Vincent Conitzer and Tuomas Sandholm. New complexity results about Nash equilibria. *Games and Economic Behavior*, 63(2):621–641, 2008. Earlier versions appeared in IJCAI-03 and as technical report CMU-CS-02-135.

[69] Vincent Conitzer and Tuomas Sandholm. Computing optimal outcomes under an expressive representation of settings with externalities. *Journal of Computer and System Sciences*, 2010. Special Issue on Knowledge Representation and Reasoning. To appear. Early version in AAAI-05.

[70] Vincent Conitzer and Tuomas Sandholm. Expressive markets for donating to charities. *Artificial Intelligence*, 2010. Special Issue on Representing, Processing, and Learning Preferences: Theoretical and Practical Challenges. To appear. Early version in EC-04.

[71] Vincent Conitzer, Tuomas Sandholm, and Jérôme Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 1–33, 2007. Early versions in AAAI-02 and TARK-03.

[72] Vincent Conitzer, Tuomas Sandholm, and Paolo Santi. Combinatorial auctions with k-wise dependent valuations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 248–254, Pittsburgh, PA, USA, 2005.

[73] Vincent Conitzer and Makoto Yokoo. Using mechanism design to prevent false-name manipulations. *AI Magazine*, 2010. Special Issue on Algorithmic Game Theory. To appear.

[74] Peter Cramton, Yoav Shoham, and Richard Steinberg. *Combinatorial Auctions*. MIT Press, 2006.

[75] Constantinos Daskalakis, Paul Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 71–78, 2006.

[76] Constantinos Daskalakis, Aranyak Mehta, and Christos H. Papadimitriou. A note on approximate Nash equilibria. In *Workshop on Internet and Network Economics (WINE)*, pages 297–306, Patras, Greece, 2006.

[77] Geoffroy de Clippel, Victor Naroditskiy, and Amy Greenwald. Destroy to save. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 207–214, Stanford, CA, USA, 2009.

[78] Yvo Desmedt and Edith Elkind. Equilibria of plurality voting with abstentions. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 347–356, Cambridge, MA, USA, 2010.

[79] John R. Douceur. The Sybil attack. In *First International Workshop on Peer-to-Peer Systems*, pages 251–260, Cambridge, MA, USA, 2002.

[80] Kousha Etessami and Mihalis Yannakakis. On the complexity of Nash equilibria and other fixed points (extended abstract). In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 113–123, 2007.

[81] Piotr Faliszewski and Ariel D. Procaccia. AI's war on manipulation: Are we winning? Working paper.

[82] Joseph Farfel and Vincent Conitzer. Aggregating value ranges: preference elicitation and truthfulness. *Autonomous Agents and Multi-Agent Systems*, 2009. To appear.

[83] Joseph Farfel and Vincent Conitzer. Turing Trade: A hybrid of a Turing test and a prediction market. In *Proceedings of the Conference on Auctions, Market Mechanisms and Their Applications (AMMA-09)*, pages 61–73, Boston, MA, USA, 2009.

[84] Tomas Feder, Hamid Nazerzadeh, and Amin Saberi. Approximating Nash equilibria using small-support strategies. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 352–354, San Diego, CA, USA, 2007.

[85] Ehud Friedgut, Gil Kalai, and Noam Nisan. Elections can be manipulated often. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 243–249, 2008.

[86] Arpita Ghosh and Mohammad Mahdian. Charity auctions on social networks. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1019–1028, 2008.

[87] Itzhak Gilboa, Ehud Kalai, and Eitan Zemel. The complexity of eliminating dominated strategies. *Mathematics of Operation Research*, 18:553–565, 1993.

[88] Itzhak Gilboa and Eitan Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.

[89] Georg Gottlob and Gianluigi Greco. On the complexity of combinatorial auctions: Structured item graphs and hypertree decompositions. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 152–161, San Diego, CA, USA, 2007.

[90] Mingyu Guo and Vincent Conitzer. Better redistribution with inefficient allocation in multi-unit auctions with unit demand. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 210–219, Chicago, IL, USA, 2008.

[91] Mingyu Guo and Vincent Conitzer. Undominated VCG redistribution mechanisms. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1039–1046, Estoril, Portugal, 2008.

[92] Mingyu Guo and Vincent Conitzer. Worst-case optimal redistribution of VCG payments in multi-unit auctions. *Games and Economic Behavior*, 67(1):69–98, 2009. Special Section of Games and Economic Behavior Dedicated to the 8th ACM Conference on Electronic Commerce.

[93] Mingyu Guo and Vincent Conitzer. Computationally feasible automated mechanism design: General approach and case studies. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1676–1679, Atlanta, GA, USA, 2010. NECTAR track.

[94] Mingyu Guo and Vincent Conitzer. False-name-proofness with bid withdrawal (extended abstract). In *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1475–1476, Toronto, Canada, 2010.

[95] Mingyu Guo and Vincent Conitzer. Optimal-in-expectation redistribution mechanisms. *Artificial Intelligence*, 174(5-6):363–381, 2010.

[96] Mingyu Guo and Vincent Conitzer. Strategy-proof allocation of multiple items between two agents without payments or priors. In *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 881–888, Toronto, Canada, 2010.

[97] Mingyu Guo, Vincent Conitzer, and Daniel M. Reeves. Competitive repeated allocation without payments. In *Proceedings of the Fifth Workshop on Internet and Network Economics (WINE)*, pages 244–255, Rome, Italy, 2009.

[98] Erik Halvorson, Vincent Conitzer, and Ronald Parr. Multi-step multi-sensor hider-seeker games. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 159–166, Pasadena, CA, USA, 2009.

[99] B. Paul Harrenstein, Mathijs M. de Weerdt, and Vincent Conitzer. A qualitative Vickrey auction. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 197–206, Stanford, CA, USA, 2009.

[100] Sergiu Hart and Yishay Mansour. The communication complexity of uncoupled Nash equilibrium procedures. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2007.

[101] Edith Hemaspaandra, Holger Spakowski, and Jörg Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, December 2005.

[102] Samuel Ieong and Yoav Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 193–292, Vancouver, BC, Canada, 2005.

[103] Atsushi Iwasaki, Vincent Conitzer, Yoshifusa Omori, Yuko Sakurai, Taiki Todo, Mingyu Guo, and Makoto Yokoo. Worst-case efficiency ratio in false-name-proof combinatorial auction mechanisms. In *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 633–640, Toronto, Canada, 2010.

[104] Manish Jain, Dmytro Korzhyk, Ondrej Vanek, Vincent Conitzer, Michal Pechoucek, and Milind Tambe. A double oracle algorithm for zero-sum security games on graphs. In *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Taipei, Taiwan, 2011.

[105] Radu Jurca. *Truthful Reputation Mechanisms for Online Systems*. PhD thesis, EPFL, 2007.

[106] Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordóñez, and Milind Tambe. Computing optimal randomized resource allocations for massive security games. In *Proceedings of the Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 689–696, Budapest, Hungary, 2009.

[107] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 805–810, Atlanta, GA, USA, 2010.

[108] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Solving Stackelberg games with uncertain observability. In *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Taipei, Taiwan, 2011.

[109] D. Lacy and E. Niou. A problem with referenda. *Journal of Theoretical Politics*, 12(1):5–31, 2000.

[110] Daniel Lehmann, Rudolf Müller, and Tuomas Sandholm. The winner determination problem. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 12, pages 297–317. MIT Press, 2006.

[111] Joshua Letchford and Vincent Conitzer. Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 83–92, Cambridge, MA, USA, 2010.

[112] Joshua Letchford, Vincent Conitzer, and Kamal Jain. An ethical game-theoretic solution concept for two-player perfect-information games. In *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, pages 696–707, Shanghai, China, 2008.

[113] Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *Proceedings of the Second Symposium on Algorithmic Game Theory (SAGT-09)*, pages 250–262, Paphos, Cyprus, 2009.

[114] Hervé Moulin. Almost budget-balanced VCG mechanisms to assign multiple objects. *Journal of Economic Theory*, 144(1):96–119, 2009.

[115] Noam Nisan. Bidding languages for combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 9, pages 215–231. MIT Press, 2006.

[116] Naoki Ohta, Vincent Conitzer, Ryo Ichimura, Yuko Sakurai, Atsushi Iwasaki, and Makoto Yokoo. Coalition structure generation utilizing compact characteristic function representations. In *Proceedings of the Fifteenth International Conference on Principles and Practice of Constraint Programming*, pages 623–638, Lisbon, Portugal, 2009.

[117] Naoki Ohta, Vincent Conitzer, Yasufumi Satoh, Atsushi Iwasaki, and Makoto Yokoo. Anonymity-proof Shapley value: Extending Shapley value for coalitional games in open environments. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 927–934, Estoril, Portugal, 2008.

[118] Naoki Ohta, Atsushi Iwasaki, Makoto Yokoo, Kohki Maruono, Vincent Conitzer, and Tuomas Sandholm. A compact representation scheme for coalitional games in open anonymous environments. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 697–702, Boston, MA, USA, 2006.

[119] Abraham Othman and Tuomas Sandholm. The cost and windfall of manipulability. In *Proceedings of the Second International Workshop on Computational Social Choice (COMSOC)*, Liverpool, England, 2008.

[120] David Parkes. Iterative combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 2, pages 41–77. MIT Press, 2006.

[121] Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 895–902, Estoril, Portugal, 2008.

[122] Steve Phelps, Peter McBurnley, Simon Parsons, and Elizabeth Sklar. Co-evolutionary auction mechanism design. *Lecture Notes in AI*, 2531, 2002.

[123] James Pita, Manish Jain, Fernando Ordóñez, Christopher Portway, Milind Tambe, and Craig Western. Using game theory for Los Angeles airport security. *AI Magazine*, 30(1):43–57, 2009.

[124] Ryan Porter, Yoav Shoham, and Moshe Tennenholtz. Fair imposition. *Journal of Economic Theory*, 118:209–228, 2004. Early version appeared in IJCAI-01.

[125] Ariel D. Procaccia and Jeffrey S. Rosenschein. Average-case tractability of manipulation in voting via the fraction of manipulators. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Honolulu, HI, USA, 2007.

[126] Ariel D. Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 177–186, Stanford, CA, USA, 2009.

[127] Tuomas Sandholm. Optimal winner determination algorithms. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 14, pages 337–368. MIT Press, 2006.

[128] Tuomas Sandholm and Craig Boutilier. Preference elicitation in combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 10, pages 233–263. MIT Press, 2006.

[129] Tuomas Sandholm, Vincent Conitzer, and Craig Boutilier. Automated design of multistage mechanisms. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1500–1506, Hyderabad, India, 2007.

[130] Tuomas Sandholm, Andrew Gilpin, and Vincent Conitzer. Mixed-integer programming methods for finding Nash equilibria. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 495–501, Pittsburgh, PA, USA, 2005.

[131] Paolo Santi, Vincent Conitzer, and Tuomas Sandholm. Towards a characterization of polynomial preference elicitation with value queries in combinatorial auctions. In *Conference on Learning Theory (COLT)*, pages 1–16, Banff, Alberta, Canada, 2004.

[132] Peng Shi, Vincent Conitzer, and Mingyu Guo. Prediction mechanisms that do not incentivize undesirable actions. In *Proceedings of the Fifth Workshop on Internet and Network Economics (WINE)*, pages 89–100, Rome, Italy, 2009.

[133] Satinder Singh, Michael Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 541–548, Stanford, CA, USA, 2000.

[134] Jason Tsai, Shyamsunder Rathi, Christopher Kiekintveld, Fernando Ordonez, and Milind Tambe. IRIS - a tool for strategic security allocation in transportation networks. In *The Eighth International Conference on Autonomous Agents and Multiagent Systems - Industry Track*, 2009.

[135] Alan Turing. Computing machinery and intelligence. *Mind*, 59, 1950.

[136] Bernhard von Stengel and Shmuel Zamir. Leadership games with convex strategy sets. *Games and Economic Behavior*, 69:446–457, 2010.

[137] Yevgeniy Vorobeychik. *Mechanism Design and Analysis Using Simulation-Based Game Models*. PhD thesis, University of Michigan, 2008.

[138] Liad Wagman and Vincent Conitzer. Optimal false-name-proof voting rules with costly voting. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 190–195, Chicago, IL, USA, 2008.

[139] Liad Wagman and Vincent Conitzer. Strategic betting for competitive agents. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 847–854, Estoril, Portugal, 2008.

[140] Justin Wolfers and Eric Zitzewitz. Prediction Markets. *The Journal of Economic Perspectives*, 18(2):107–126, 2004.

[141] Lirong Xia and Vincent Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 196–201, Chicago, IL, USA, 2008.

[142] Lirong Xia and Vincent Conitzer. Generalized scoring rules and the frequency of coalitional manipulability. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 109–118, Chicago, IL, USA, 2008.

[143] Lirong Xia and Vincent Conitzer. A sufficient condition for voting rules to be frequently manipulable. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 99–108, Chicago, IL, USA, 2008.

[144] Lirong Xia and Vincent Conitzer. Finite local consistency characterizes generalized scoring rules. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 336–341, Pasadena, CA, USA, 2009.

[145] Lirong Xia and Vincent Conitzer. Compilation complexity of common voting rules. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 915–920, Atlanta, GA, USA, 2010.

[146] Lirong Xia and Vincent Conitzer. Stackelberg voting games: Computational aspects and paradoxes. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 921–926, Atlanta, GA, USA, 2010.

[147] Lirong Xia and Vincent Conitzer. Strategy-proof voting rules over multi-issue domains with restricted preferences. In *Proceedings of the Sixth Workshop on Internet and Network Economics (WINE)*, Stanford, CA, USA, 2010.

[148] Lirong Xia, Vincent Conitzer, and Jérôme Lang. Voting on multiattribute domains with cyclic preferential dependencies. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 202–207, Chicago, IL, USA, 2008.

[149] Lirong Xia, Vincent Conitzer, and Jérôme Lang. Aggregating preferences in multi-issue domains by using maximum likelihood estimators. In *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 399–406, Toronto, Canada, 2010.

[150] Lirong Xia, Vincent Conitzer, and Ariel D. Procaccia. A scheduling approach to coalitional manipulation. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 275–284, Cambridge, MA, USA, 2010.

[151] Lirong Xia, Michael Zuckerman, Ariel D. Procaccia, Vincent Conitzer, and Jeffrey Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353, Pasadena, CA, USA, 2009.

[152] Zhengyu Yin, Dmytro Korzhyk, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness. In *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1139–1146, Toronto, Canada, 2010.

[153] Makoto Yokoo, Vincent Conitzer, Tuomas Sandholm, Naoki Ohta, and Atsushi Iwasaki. Coalitional games in open anonymous environments. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 509–514, Pittsburgh, PA, USA, 2005.

[154] Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. Robust combinatorial auction protocol against false-name bids. *Artificial Intelligence*, 130(2):167–181, 2001.

[155] Makoto Yokoo, Yuko Sakurai, and Shigeo Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. *Games and Economic Behavior*, 46(1):174–188, 2004.

[156] H. Peyton Young. Optimal voting rules. *Journal of Economic Perspectives*, 9(1):51–64, 1995.

[157] H. Peyton Young and Arthur Levenglick. A consistent extension of Condorcet's election principle. *SIAM Journal of Applied Mathematics*, 35(2):285–300, 1978.

[158] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. SybilLimit: A near-optimal social network defense against sybil attacks. *IEEE/ACM Transactions on Networking (ToN)*, 18(3):885–898, 2010.

[159] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. SybilGuard: Defending against sybil attacks via social networks. *IEEE/ACM Transactions on Networking (ToN)*, 16(3):576–589, 2008.

[160] Michael Zuckerman, Ariel D. Procaccia, and Jeffrey S. Rosenschein. Algorithms for the coalitional manipulation problem. *Artificial Intelligence*, 173(2):392–412, 2009.