1 Research interests and style

My research focuses on issues in the intersection of computer science (especially artificial intelligence) and economics. This includes the design of new marketplaces and other negotiation protocols (both for humans and for software agents) that allow agents to express their preferences naturally and accurately, and that generate good outcomes based on these preferences. It also includes the design of agents that can act strategically in settings where multiple agents all pursue their own interests. This requires us to use concepts from game theory, as well as to operationalize these concepts by finding efficient algorithms for computing the corresponding solutions. Finally, my research includes the study of all settings in computer science in which multiple parties will act in their own self-interest, as well as the design of incentive mechanisms to reach good outcomes in spite of such behavior.

I enjoy taking a creative approach to these issues, attempting to find new ways in which computer science can improve the welfare of humanity. While I have made a few significant contributions to previously recognized important problems, most of my research consists of creating a new approach to a high-level problem and finding the right formalization for this approach. I try to distill the essential computational problem and to deeply understand this problem, at which point the proper way to attack it algorithmically usually becomes clear. I find this to be an immensely rewarding process, and I believe that the results have tremendous potential to improve the world.

2 Brief overview of research

In this statement, I will discuss only my main areas of research, and not results that I have obtained in other areas. The description will also remain at a high level, omitting most of the detail on individual papers and results. For more detail, please see the long version of my research statement (available online at http://www.cs.cmu.edu/~conitzer/long_research.pdf).

2.1 Expressive preference aggregation

Traditionally, marketplaces and other negotiation protocols have forced their participants to express their preferences in very simplified and approximate fashions. For example, in auctions, bidders have had to place separate bids on individual items, even when how much they valued one item depended on whether they won another item. Recently, there has been a shift towards designing protocols that allow agents to express their preferences in their full complexity. For example, in a combinatorial auction, there are multiple items for sale, and bidders are allowed to place bids on sets of items. Such expressivity can greatly improve the efficiency of the final allocation, but it also creates several problems. First, the optimization problem of choosing the best outcome under the reported preferences typically becomes hard. Second, it is generally not feasible for an agent to communicate all of its preferences, so the protocol has to selectively elicit only the information that it needs to choose the right outcome.

Combinatorial auctions (and generalizations such as combinatorial exchanges) are the most-studied example of expressive preference aggregation, and I have obtained a number of significant results on this topic. Specifically, I have studied various ways in which the preferences of the bidders can be restricted to simplify the problems above (while maintaining the benefits of expressiveness). In addition, as a consultant for CombineNet, Inc., a company
that focuses mainly on providing optimal solutions for strategic sourcing, I have a significant amount of experience in how real-world combinatorial auctions are created and run.

I have also designed expressive negotiation protocols for entirely different settings, to which nobody had previously thought to apply the paradigm of expressive negotiation. For example, when donating money to a charity, it is possible to use the contemplated donation as negotiating material to induce other parties interested in the charity to donate more. Such negotiation is sometimes done through simple matching offers, but these allow for only limited negotiation. I created a language for expressing very general types of matching offers over multiple charities. At a 2005 workshop in Dagstuhl, I used this approach to raise money for tsunami victims; I also used it to raise money for victims of Hurricane Katrina over the web. I also introduced a more general framework for negotiation over actions that affect the utilities of agents that are not otherwise involved (for instance, a company reducing its pollution levels). In both settings, I analyzed the complexity of choosing the optimal outcome given the offers made by the agents, and designed algorithms for solving this problem.

A general approach to aggregating the preferences of multiple agents is to vote over the possible outcomes. In the most general framework, each voter ranks all the alternatives, and a voting rule maps these rankings into a single aggregate ranking. It turns out that some important voting rules are in fact NP-hard to execute. I have created several algorithmic techniques that allow us to execute these rules on much larger elections than could previously be solved. For the most common voting rules, I have designed algorithms that elicit enough information from the voters to determine the winning outcome, and proven the optimality of these algorithms by giving a lower bound on the communication complexity of these rules. I have also shown that some voting rules can be interpreted as maximum likelihood estimators of the “correct” outcome (and other rules cannot).

2.2 Computing game-theoretic solutions

Game theory studies how agents should act when their utilities are affected by the actions of other agents, who pursue their own interests. In such games, what the optimal action for each agent is generally depends on the actions of the other agents, so that it is not immediately obvious how to define what the “right” actions to play are. Game theory provides a number of such definitions (also known as solution concepts). For example, one strategy is said to be dominated if there is another strategy that performs better against any opponent strategies. As another example, a Nash equilibrium prescribes a strategy for each agent, in such a way that if all players play their assigned strategies, no single player has an incentive to deviate. The study of how to compute strategies consistent with these (and other) solution concepts had received only limited interest until recently. Knowing how to compute game-theoretic strategies has important applications to artificial intelligence and the design of software agents (including, for example, computer poker players, trading agents, soccer-playing robots, sensor networks that are used for surveillance, etc.). It is also important for anticipating how agents will behave in complex multiagent protocols (such as those described in the previous section).

I have derived a number of fundamental results on the complexity of computing game-theoretic solutions, including the computation of dominated strategies and of Nash equilibria. (The results on the complexity of computing Nash equilibria are especially well-known and the paper appears on the reading list of many courses on computational game theory.) I have also created a variety of techniques to compute (or to help in computing) Nash equilibria. One of these techniques is interesting in and of itself because it actually defines a spectrum of solution concepts, with dominance on the one end of the spectrum, Nash equilibrium on the other, and new concepts in-between. I have also derived complexity results and algorithms for a number of other solution concepts, including optimal strategies when early commitment is possible, the core, and the Shapley value. In addition, I have derived a number of results on how agents can learn to play a game, in settings where the game is not fully known or where the opponent does not necessarily play optimally. As an example of how to solve a specific game, I recently designed an optimal computer player for (one variant of) the game of Liar’s Dice.

2.3 Designing the rules of the game (mechanism design)

Aggregating the preferences of self-interested agents is complicated by the fact that they will strategically misreport their preferences if this is to their benefit. Thus, we may end up choosing an outcome that is desirable with respect to the agents’ reported preferences, but undesirable with respect to their true preferences. Mechanism
design is the study of how to aggregate preferences in such a way that desirable outcomes (with respect to the true preferences) are obtained in spite of such strategic behavior. The predominant approach to doing so is to design truthful mechanisms, that is, to choose outcomes in such a way that no agent has an incentive to misreport. There are a few well-known general-purpose mechanisms, such as the Vickrey-Clarke-Groves (VCG) mechanism, which obtains truthfulness by specifying payments that the agents should make. However, each of these mechanisms has its drawbacks. For example, the VCG mechanism is known to be vulnerable to collusion, and I have done work showing that this vulnerability is especially extreme in combinatorial auctions and exchanges.

I decided to take a different approach, and look at mechanism design as a computational problem. Rather than try to come up with a general formula for engineering incentives across a large range of settings, I studied how the optimal incentive mechanism can be automatically computed when the specific setting is given. I analyzed the complexity of this problem and created basic algorithms for solving it. We have since turned this into a major line of research called automated mechanism design, and my advisor recently obtained a US$ 1,100,000 NSF ITR grant to continue research on automated mechanism design. We have also used automated mechanism design on a couple of industrial projects.

Another interesting observation is that agents may not always have the computational resources to find the action that is in their own best interest. Mechanism design has traditionally made the conservative assumption that agents will find this action, and this is in fact what motivates the focus on truthful mechanisms: by a result known as the revelation principle, under this assumption, it is impossible to obtain results that are better than those of the best truthful mechanism. However, I have proven results that show that the revelation principle falls apart without this assumption. For example, in some settings, if agents cannot always solve NP-hard problems, then better results can be obtained using mechanisms that are not truthful. Also, in elections, finding beneficial misreports is computationally hard under certain voting rules. I have shown how to modify most voting rules to make manipulating elections in this manner especially hard (even when the original voting rule is easy to manipulate), as well as that for many voting rules, certain types of manipulation are hard even when there are very few possible outcomes. Finally, I have suggested a new approach to mechanism design under which mechanisms are incrementally made more truthful (but not necessarily completely truthful).

3 Future research

I am thrilled by the potential of my research area. We have only begun to see how techniques from computer science (and artificial intelligence in particular) can be used to drastically improve the efficiency of marketplaces and other negotiation protocols. As a consultant for CombineNet, Inc., I am continuously on the forefront of the integration of these techniques into industry, and it never ceases to amaze me how much potential is yet left untapped. I am especially excited about, and wish to pursue, the design of expressive negotiation protocols for settings involving public goods (i.e., goods that can benefit large populations, such as clean air, national defense, and pure research) and, closely related, externalities (effects that one agent’s actions have on an otherwise unrelated agent). Arguably, most of the major problems facing the world today concern such settings. This is closely related to my work in social choice theory (voting), which I hope to continue as well. It may help create new (e.g., web-based) methods for citizens to become directly involved in decisions that are currently left to the government.

I also remain very excited about the design of algorithms for reasoning strategically. Here, many open questions remain, not only on how hard it is to compute various game-theoretic solution concepts, but also on which (if any) of these concepts is the best one to use in any given setting. These are topics of tremendous interest for artificial intelligence, with applications to the design of computer players for game-theoretically nontrivial games, as well as to a variety of real-world applications.

Additionally, there are many open questions in mechanism design that I wish to explore. Automated mechanism design does not yet scale to very large instances, and I believe that significant breakthroughs based on the structure of the problem must be made to improve this. Equally significantly, I believe that the observation that computationally bounded agents may not act in a strategically optimal way turns the field of mechanism design on its head, but we still do not have a good general framework for taking advantage of this observation.

Finally, I hope that I will be fortunate enough to continue to generate ideas for entirely new research topics within my area. I have several new ideas that I hope to be able to crystallize soon.