13

Conclusion and Future Directions

In recent years, rapid developments in computers and networks have brought big changes to human society. These changes have led to many new interdisciplinary areas among which the interdisciplinary area lying in the intersection of Computer Science and Economics has attracted much attention. Computational Social Choice is a burgeoning subarea in this intersection. This dissertation includes my Ph.D. research on two aspects of Computational Voting Theory, which is the most active and major branch of Computational Social Choice. The novel research contributions are as follows.

1. **Game-theoretic aspects** (Chapter 4–Chapter 7). In this part I examined the motivation and possibility to circumvent the Gibbard-Satterthwaite theorem by using computational complexity as a barrier against voters' strategic behavior.

2. **Combinatorial aspects** (Chapter 9–Chapter 12). In this part I focused on the design and analysis of computationally tractable voting rules for multi-issue domains to overcome the computational difficulties in preference representation and aggregation caused by the exponential blowup in the number of alterna-

tives.

13.1 Summary of Chapters

Chapter 1 served as a general and high-level introduction to the work included in this dissertation, where I briefly described the motivating questions for my research, the methodology we have developed and the results we have obtained, and how these results answered the motivating questions. Chapter 2 introduced notation used throughout the dissertation, definitions of some common voting rules and some axiomatic properties for voting, and gave a brief introduction to other major research directions in Computational Social Choice. Chapter 3 is a mixture of introduction and preliminaries for the game-theoretic aspects of my work, which are covered in Chapter 4 through Chapter 7.

In Chapter 4, we characterized the computational complexity for the unweighted coalitional manipulation problem for three common voting rules. We showed that UCM is NP-complete for maximin (Section 4.1) and ranked pairs (Section 4.2), and in P for Bucklin (Section 4.3). These worst-case hardness results imply that at least for maximin and ranked pairs, computational complexity can provide some protection against manipulation. Therefore, for these results, it gives an affirmative answer to the question "Can we use computational complexity to prevent manipulation?"

In Chapter 5, we continued investigating the possibility of using computational complexity as a barrier against manipulation. We focused on the question "Is computational complexity a *strong* barrier against manipulation?" Unfortunately, the answer was quite negative, as illustrated by our research in two directions. In Section 5.2, we pursued the "frequency of manipulability" approach, and showed that for most common voting rules, with a high probability the UCM problem is computationally trivial. To prove this result, we introduced generalized scoring rules, and then characterized the frequency of manipulability for all generalized scoring rules.

To show how general this class of voting rules is, we gave a concise axiomatic characterization of it in Section 5.3. In Section 5.4, we pursued an approximation approach. We devised an efficient polynomial-time algorithm that approximately computes the smallest number of manipulators that are needed to make a given alternative win, for all positional scoring rules.

Since computational complexity does not seem to be a strong barrier against manipulation, we need to look for other ways to circumvent Gibbard-Satterthwaite. In Chapter 6, we examined some preliminary ideas about preventing manipulations by restricting the manipulator's information about the other voters' votes. Our results are encouraging: restricting the manipulator's preferences can make a certain type of manipulation, which we called "dominating manipulation", computationally hard or even impossible.

In fact, the very first question that should be asked is probably not "How can we circumvent Gibbard-Satterthwaite?", but is rather, "Is the strategic behavior undesirable?" Surprisingly, in the literature little work attempted to answer this question. The difficulty mainly comes from the fact that there are too many (trivial) equilibria in voting games. In Chapter 7, we partly answered this question by showing that in any Stackelberg voting game, there is a unique winner across all equilibria, and it is sometimes ranked within the bottom two positions in all voters' true preferences, with only a few exceptions. Therefore, the main theoretical results of Chapter 7 (the paradoxes) are extremely negative. Their high-level message is what we may have expected to see: sometimes the strategic behavior of the voters leads to extremely undesirable outcomes. This justifies the previous line of research of using computational complexity to prevent manipulation. We also devised some techniques to speed up the computation of the equilibrium outcome. These techniques were used in our simulations, which showed that, surprisingly, on average the equilibrium outcome is preferred by slightly more voters compared to the winner where the voters vote truthfully.

The combinatorial voting part of this thesis started in Chapter 8, where we introduced the notation for combinatorial voting, multiple-election paradoxes, CP-nets, sequential voting rules, and important criteria for designing new voting rules. We also evaluated voting rules proposed in previous work by these criteria (Table 8.1). We observed that all previous approaches either used voting languages that lack expressivity, or are computationally intractable. This motivated my work in Chapter 9 and Chapter 10.

Chapter 9 and Chapter 10 both focused on designing new voting methods for combinatorial voting. We first showed quantitatively in Chapter 9 that (possibly cyclic) CP-nets are much more usable than the voting languages used in sequential voting and issue-by-issue voting. The voting methods we proposed in Chapter 9 and Chapter 10 both allow voters to use (possibly cyclic) CP-nets to represent their preferences. In the framework we proposed in Chapter 9, which we called H-compositions, we first consider an induced graph over all alternatives by applying local voting rules, then apply a choice set function to select the winner. We showed that H-compositions are an extension of sequential voting rules, and then examined whether they satisfy some common axiomatic properties. We also studied how to compute the winners for the H-compositions for a common choice set function called the Schwartz set.

In Chapter 10, we took an MLE approach by extending Condorcet's MLE model to multi-issue domains. We studied the relationship between the voting correspondences defined by the MLE approach and sequential voting correspondences, and showed that the MLE approach gives us genuinely new correspondences. We then focused on multi-issue domains composed of binary issues; for these, we proposed a general family of distance-based noise models that are parameterized by a threshold. We identified the complexity of winner determination for the corresponding MLE voting rules in the two most important subcases of this framework.

Chapter 11 and Chapter 12 were both devoted to studying game-theoretic aspects of combinatorial voting. Chapter 11 in some sense told us the same high-level message as Chapter 7, which is: strategic behavior of the voters sometimes leads to extremely undesirable outcomes. More precisely, we studied strategic sequential voting, which is a complete-information extensive-form game of sequential voting in multi-issue domains. We focused on domains characterized by multiple binary issues, and illustrated three types of multiple-election paradoxes in strategic sequential voting. We showed that changing the order of the issues cannot completely prevent such paradoxes. We also investigated the possibility of avoiding the paradoxes for strategic sequential voting by imposing some constraints on the profile.

Finally, Chapter 12 pursued an older line of research to circumvent Gibbard-Satterthwaite, which has typically been pursued by economists. We studied how to restrict voters' preferences over multi-issue domains to obtain strategy-proof voting rules. Our main result is a simple full characterization of strategy-proof voting rules over restricted sets of lexicographic profiles. This result is a counterpart of a well-known previous characterization of strategy-proof voting rules over restricted sets of separable profiles by Le Breton and Sen (1999).

13.2 Future Directions

Computational Social Choice is still in its infancy. There are many promising theoretical and practical directions for future research. On the one hand, I plan to further explore the conceptual changes in Social Choice brought by computational thinking. On the other hand, I plan to work on designing and employing new voting systems for preference representation and aggregation in Multi-Agent Systems, which is one of the best application fields for Computational Social Choice. In what follows, I will point out some future/on-going research directions for the game-theoretic aspects and combinatorial aspects of Computational Voting Theory, respectively.

13.2.1 Game-Theoretic Aspects

The computational complexity of UCM has been resolved for many common voting rules (see Table 3.1). It can be easily observed from the table that multi-stage voting rules seem to be harder to manipulate. In fact, as far as we know, the only four voting rules for which UCM is hard for only one manipulator are all composed of multiple stages (STV, ranked pairs, Nanson's and Baldwin's rules). Among them, STV, Nanson's and Baldwin's rules are defined in a very similar way: in each round, a voting correspondence is used to eliminate some alternatives based on some "scores" (plurality score for STV, Borda score for Nanson's and Baldwin's rules). We note that for plurality and Borda, manipulation is easy for one manipulator. Hence, it seems that the multi-stage-elimination pattern used in STV, Nanson's and Baldwin's rules is an effective way to make manipulation computationally hard. Therefore, we may ask the following open question: Can we characterize the computational complexity of UCM for the voting rules that are defined similar by STV, Nanson's and Baldwin's rules? For example, we can study the computational complexity for UCM under the voting rules where a positional scoring rule that is different from plurality and Borda is applied in each round, and the alternatives whose scores are the lowest (or below the average score) are eliminated.

There are also some open questions about the "typical-case" complexity of manipulation. Recall that our characterization of the frequency of manipulability for generalized scoring rules (Section 5.2) leaves a knife-edge case open, which is the case where the number of manipulators is $\Theta(\sqrt{n})$. Thus, the "typical-case" complexity of manipulation is an open question for such cases. Another important assumption we made in our characterization is that the voters' votes are i.i.d. However, in real life the voters' votes are generally correlated. Therefore, it is interesting to investigate the frequency of manipulability with correlated voters. We note that Walsh (2009) studied both questions by simulations.

Open questions along the research direction of approximating the UCO problem include extending our scheduling approach to other common voting rules, for example, generalized scoring rules. Recently, Zuckerman et al. (2011) proposed a $(\frac{5}{2})$-approximation algorithm for UCO under maximin. It would be nice to see a unified approach for a large class of voting rules.

In Chapter 6, we took a first step in the research direction of using information constraints to make manipulations computationally hard or even impossible. There are many interesting open questions left for future research. Our results showed that by restricting the manipulator's information, sometimes we can increase the hardness of computing dominating manipulations from in P to NP-hard. One open question here is to characterize the exact computational complexity of computing dominating manipulations under information constraints. We could analyze the "typical-case" complexity, or it might be possible to prove completeness results for higher levels of the polynomial hierarchy. Since we only studied manipulation with one manipulator in Chapter 6, we may also consider using information constraints to prevent other types of strategic behavior in our framework, including coalitional manipulation, bribery, and control, or even more generally, to prevent strategic behavior in other mechanism design or game-theoretic settings. Also, the notion of dominating manipulation might be too strong, in the sense that it corresponds to a very cautious manipulator who always wants to make sure that whatever the possible world is, she is never worse-off (and sometimes better-off). This does not model some real-life situations, where manipulators may want to take some risk to obtain higher payoffs. One important next step is to investigate other types of manipulation when the manipulators have incomplete information.

In addition to coalitional manipulation, bribery and control, some other models of voters' strategic behavior have been studied in Computational Social Choice. For example, *false-name manipulation* (Yokoo et al., 2004) refers to the strategic behavior of an agent who creates multiple false identities to participate in auctions or elections, to make the outcome more preferable to her. See (Conitzer and Yokoo, 2010) for an overview. In this voting setting, this problem is related to a special control type called "control by adding new voters" (Bartholdi et al., 1992). Because traditional manipulation is a special case of false-name manipulation, it is not surprising to see negative results in the voting setting. In fact, Conitzer (2008) gave a complete characterization of randomized false-name-proof voting rules that satisfy voluntary participation. The characterization is significantly more negative than the characterization of randomized strategy-proof voting rules obtained by Gibbard (1977). Some positive results have also been obtained to prevent false-name manipulations. Wagman and Conitzer (2008) modeled the cost of creating false identities, and designed optimal false-name-proof voting rules for two alternatives. Conitzer et al. (2010) proposed a voting rule that uses the social-network structure of the voters to detect potential false identities, and then block them from casting votes. I believe that designing new ways to protect elections from false-name-manipulations deserves more attention, and again, we may consider using information constraints to prevent false-name manipulation.

Another example of voters' strategic behavior is *safe manipulation* (Slinko and White, 2008). In the safe-manipulation model, a manipulator (the *leader*) can send a message to all voters who have the same preferences as her (the *followers*), asking them to cast the same vote V which is not necessarily the same as their true preferences. If there exists such a vote V that (1) no matter how many followers follow the suggestion of the leader, they are never worse off, meaning that the winner is at least as preferred as the winner when all voters report their preferences truthfully, and (2) sometimes they are strictly better-off, then this is a safe manipulation.

Slinko and White (2008) extended the Gibbard-Satterthwaite result to the notion of safe manipulation. Therefore, we can ask the question "Is computational complexity a barrier against safe manipulation?" In fact, the computational complexity of safe manipulation has been investigated for some common voting rules (Hazon and Elkind, 2010; Ianovski et al., 2011), but it is still open for some other common voting rules, for example, positional scoring rules in general. Once again, we can ask the "worst-case" vs. "typical-case" question, and see to what extent restricting manipulators' information about the preferences of the other voters (for example, the maximum number of followers) can help prevent safe manipulations. At a high level, it is still not clear how well the safe-manipulation model captures the voters' strategic behavior in coalition formation. In this dissertation we only studied the case where there is only one group of manipulators. In real life, sometimes there are multiple groups of manipulators aiming at making different alternatives win. Also, the group of manipulators were given exogenously. Therefore, it would be nice to have some justifications or improvements of the coalitional manipulation model. For example, Bachrach et al. (2011) modeled the coalition formation process of the manipulators as a coalitional game, and investigated its computational aspects.

Modeling a voting process as a game and analyzing its equilibrium outcomes is an old yet fascinating topic. In the Stackelberg voting games studied in Chapter 7, we assumed that the voters vote according to an exogenously-given order, and every voter cast exactly one vote. However, in many online rating systems, a voter is free to decide when she cast the vote, or simply not casting any vote. Desmedt and Elkind (2010) allowed voter to absent, but if a voter decides to absent, then she cannot come back to vote later. Therefore, the equilibrium analysis of voting games where voters can decide when to cast votes is an interesting line of research. For Stackelberg voting games, we still do not how to characterize the computational complexity of computing the SPNE outcome. We conjecture that it is PSPACE-complete (Desmedt and Elkind (2010) also proposed the same conjecture for their model). We recall that our simulation results showed that the equilibrium outcome seems to be preferred by more voters than the truthful outcome when the voters' preferences are generated i.i.d. uniformly. One open question here is: Can we obtain a theoretical result? It is also very interesting to know which voters in Stackelberg voting games have more power to control the outcome: the voters who vote early, late, or in the middle?

13.2.2 Combinatorial Aspects

Combinatorial voting settings, in which the space of all alternatives is exponential in size, constitute an important area in which techniques from Computer Science can be fruitfully applied. As we summarized in Table 8.1, none of the previous approaches to combinatorial voting (including ours) are perfect. Designing a "good" voting rule over combinatorial domains that uses a very expressive and compact language seems too ambitious to be possible. Therefore, I believe that the future design of voting rules for combinatorial domains should focus on achieving a balance among the criteria we proposed in Chapter 8, that is, the compactness and expressiveness of the voting language, and the quality (including computational efficiency) of the voting rule. Such a balance can be envisioned in the following three directions.

1. **Exploring richer connections between combinatorial voting and combinatorial auctions.** Combinatorial voting and combinatorial auctions share many common high-level characteristics: (1) Mathematically, the objectives are to decide the value of multiple variables based on participants' (cardinal or ordinal) preferences. In combinatorial auctions, one item corresponds to one variable, whose value determines which participant obtains the item. (2) The main difficulty comes from the exponential blow-up of the problem size. (3) So far, the main research agendas are proposing compact and expressive languages for the participants to express their preferences, and designing computation-