3

Introduction to Game-theoretic Aspects of Voting

In this dissertation I will discuss two directions for the game-theoretic aspects of voting. The first direction (Section 1.3.1, the left branch in Figure 1.2) aims at investigating possibilities of using computational complexity as a barrier against manipulation. The second direction (Section 1.3.2, the right branch in Figure 1.2) aims at analyzing voting games and their equilibrium outcomes. The following two sections are devoted to these two directions, respectively.

3.1 Coalitional Manipulation Problems

How to use computational complexity to escape from the Gibbard-Satterthwaite theorem (Gibbard, 1973; Satterthwaite, 1975) has attracted a lot of attention from researchers in both the Artificial Intelligence and Theoretical Computer Science communities. We recall that the main idea is, even though a manipulation ubiquitously exists, it might be computationally costly for a potential manipulator to find it. Hence, if we can prove that finding a manipulation is hard, then a potential manipulator might have less incentive to even try to look for the manipulation, and even if she does, she may not find it. In the agenda of using computational complexity as a barrier against manipulation, the main questions are the following.

1. For which voting rules can computational complexity serve as a barrier?

2. Is computational complexity a strong barrier?

To answer the first question, early work (Bartholdi et al., 1989a; Bartholdi and Orlin, 1991) has shown that when the number of candidates is not bounded, the second-order Copeland and STV rules are NP-hard to manipulate, even by a single voter. More recent research has studied how to modify other existing rules to make them hard to manipulate by a single voter (Conitzer and Sandholm, 2003; Elkind and Lipmaa, 2005).

A more general manipulation setting is that of *weighted coalitional manipulation (WCM)*. In this setting, multiple manipulators have formed a coalition, with the goal of making an agreed-upon alternative win the election. Furthermore, the voters in this setting are weighted, that is, a voter with weight k is equivalent to k unweighted voters that cast identical ballots. Weights are common, e.g., in corporate elections, where voters are weighted according to the amount of stock they hold, or Electoral College. All common voting rules studied in this paper can be easily extended to the setting where voters are weighted. (We have already seen the definition for positional scoring rules in Section 2.1.)

Definition 3.1.1. *The* Weighted Coalitional Manipulation (WCM) *problem is defined as follows. An instance is a tuple* $(r, P^{NM}, \vec{w}^{NM}, c, k, \vec{w}^M)$, *where* r *is a voting rule,* P^{NM} *is the non-manipulators' profile,* $\vec{w}^{NM}$ *represents the weights of* P^{NM}, c *is the alternative preferred by the manipulators,* k *is the number of manipulators, and* $\vec{w}^M = (w_1, \ldots, w_k)$ *represents the weights of the manipulators. We are asked whether there exists a profile* P^M *of votes for the manipulators such that* $r((P^{NM}, P^M), (\vec{w}^{NM}, \vec{w}^M)) = c$.

Conitzer et al. (2007) showed that the WCM problem is computationally hard for a variety of prominent voting rules, even when the number of alternatives is constant. Subsequent work by Hemaspaandra and Hemaspaandra (2007) dealt with positional scoring rules. They established a dichotomy theorem for the weighted coalitional manipulation problem in scoring rules: it is either NP-complete or in P, which can be easily told from the score vector $\vec{s}_m$ (see Section 2.1 for the definition of positional scoring rules). Coleman and Teague (2007) showed that WCM for the Baldwin rule is NP-hard.

A special case of weighted coalitional manipulation is its unweighted version—*unweighted coalitional manipulation (UCM)*, which is perhaps more natural in most settings (e.g., political elections). Chapter 4 studies the computational complexity of UCM for some common voting rules.

Definition 3.1.2. *The* Unweighted Coalitional Manipulation (UCM) *problem is defined as follows. An instance is a tuple* (r, P^{NM}, c, n')*, where* r *is a voting rule,* P^{NM} *is the non-manipulators' profile,* c *is the candidate preferred by the manipulators, and* n' *is the number of manipulators. We are asked whether there exists a profile* P^M *for the manipulators such that* $|P^M| = n'$ *and* $r(P^{NM} \cup P^M) = c$*.*

Progress on the UCM problem has been significantly slower than on other variations, but many of the questions have recently been resolved. The exact complexity of the problem has been investigated for some common voting rules (Faliszewski et al., 2008; Zuckerman et al., 2009; Faliszewski et al., 2010a; Narodytska et al., 2011). We will see in Chapter 4 that UCM is an NP-complete problem for some other common voting rules, i.e., maximin and ranked pairs, but is in P for Bucklin.[1] In (Xia et al., 2010), we showed that UCM is an NP-complete problem for a class of positional

[1] These results were published in Xia et al. (2009).

scoring rules (not including Borda).[2] Subsequent work (Davies et al., 2011; Betzler et al., 2011) proved that UCM is also NP-complete for Borda. Obraztsova et al. (2011); Obraztsova and Elkind (2011) investigated the computational complexity of UCM with one manipulator, for common voting rules with randomized tie-breaking.

Table 3.1: Computational complexity of UCM for common voting rules.

Voting rule	**One manipulator**		**At least two manipulators**	
Copeland	P	Bartholdi et al. (1989a)	NP-C	Faliszewski et al. (2008) Faliszewski et al. (2010a)
STV	NP-C	Bartholdi and Orlin (1991)	NP-C	Bartholdi and Orlin (1991)
Veto	P	Bartholdi et al. (1989a)	P	Zuckerman et al. (2009)
Plurality w/ Runoff	P	Zuckerman et al. (2009)	P	Zuckerman et al. (2009)
Cup	P	Conitzer et al. (2007)	P	Conitzer et al. (2007)
Maximin	P	Bartholdi et al. (1989a)	NP-C	Section 4.1
Ranked pairs	NP-C	Section 4.2	NP-C	Section 4.2
Bucklin	P	Section 4.3	P	Section 4.3
Borda	P	Bartholdi et al. (1989a)	NP-C	Davies et al. (2011) Betzler et al. (2011)
Nanson's rule	NP-C	Narodytska et al. (2011)	NP-C	Same technique as in Narodytska et al. (2011)
Baldwin's rule	NP-C	Narodytska et al. (2011)	NP-C	Same technique as in Narodytska et al. (2011)

However, all of these hardness results are worst-case results. That is, they suggest that any algorithm will require superpolynomial time to solve *some* instances. Therefore, it is natural to ask the second question: is computational complexity a strong barrier in "typical" elections? Unfortunately, several recent results seem to suggest that indeed, in various senses, hard instances of the manipulation problem are the exception rather than the rule. One type of evidence consists of "quantitative" versions of the Gibbard-Satterthwaite theorem (Friedgut et al., 2008; Dobzinski and Procaccia, 2008; Xia and Conitzer, 2008c; Isaksson et al., 2010), which state that (informally) for many voting rules, the proportion of the profiles that are manipulable is non-negligible. These results imply that there the trivial algorithm that first chooses a profile uniformly at random and then chooses a manipulator and her

[2] This result will not be further discussed in this dissertation.

false vote uniformly at random will find a manipulation instance with non-negligible probability.

Procaccia and Rosenschein (2007b) took a different perspective. They showed that for positional scoring rules, manipulation is always easy to find w.r.t. a specific *junta distribution* over the profile, which means that for many other plausible distributions, manipulation are always easy to find. Conitzer and Sandholm (2006) showed that it is impossible to design a voting rule for which manipulation is usually hard to find, if the voting rule satisfies some natural properties.

Peleg (1979), Baharad and Neeman (2002), Slinko (2002), and Slinko (2004) studied the asymptotic value of the *frequency of manipulability*, that is, the probability that a coalition of manipulators can succeed. They showed that for positional scoring rules and WMG-based voting rules, when the votes are drawn i.i.d. uniformly at random from the set of all linear orders, then the probability that a coalition of $o(\sqrt{n})$ manipulators, where n is the number of voters, can change the outcome of the election goes to 0 as n goes to infinity. More recently, Procaccia and Rosenschein (2007a) showed that for positional scoring rules, when the non-manipulators' votes are drawn i.i.d. according to some distribution that satisfies some conditions, if the number of manipulators is $o(\sqrt{n})$, then the probability that the manipulators can succeed goes to 0 as n goes to infinity; if the number of manipulator is $\omega(\sqrt{n})$, then the probability that the manipulators can succeed goes to 1.

The "dichotomy" theorem proved by Procaccia and Rosenschein (2007a) will be significantly generalized in Chapter 5. We will introduce a notion called *generalized scoring rules*, which is a type of voting rules that include almost all common voting rules.[3] In Section 5.3 we give a concise axiomatization of generalized scoring rules to show how general this class is.[4] We will show that the "dichotomy" theo-

[3] Published in Xia and Conitzer (2008b).

[4] Published in Xia and Conitzer (2009).

rem proved by Procaccia and Rosenschein (2007a) actually holds for all generalized scoring rules. Therefore, it leaves only a knife-edge case open—the case where the number of manipulators is $\Theta(\sqrt{n})$. For these cases, Walsh (2009) conducted simulation studies for the veto rule with weighted voters, and showed an interesting smooth phase-transition phenomenon. The manipulability of STV has been also studied by simulations (Walsh, 2010).

Viewing the question from yet another angle, Zuckerman et al. (2009) observed that the unweighted coalitional manipulation setting admits an optimization problem which they called *unweighted coalitional optimization (UCO)*. The goal is to find the minimum number of manipulators required to make a given candidate win the election.

Definition 3.1.3. *The* Unweighted Coalitional Optimization (UCO) *problem is defined as follows. An instance is a tuple (r, P^{NM}, c), where r is a voting rule, P^{NM} is the non-manipulators' profile, and c is the candidate preferred by the manipulators. We must find the minimum k such that there exists a set of manipulators M with $|M| = k$, and a profile P^M, that satisfies $r(P^{NM} \cup P^M) = \{c\}$.*

Zuckerman et al. (2009) gave a 2-approximation algorithm for this problem under maximin (even though this problem was not previously known to be NP-hard), and an algorithm for Borda that finds an optimal solution up to an additive term of one. More recently, Zuckerman et al. (2011) proposed an approximation algorithm for UCO for maximin.

In Chapter 5, we will present an approximation algorithm for UCO for all positional scoring rules, with an additive error bounded by m (the number of alternatives).[5] The algorithm exploits a novel connection between UCO and a specific scheduling problem. We first convert the UCO instance to a scheduling instance, then

[5] Published in Xia et al. (2010).

apply an algorithm for scheduling problems, and finally use a rounding technique to obtain a solution to the UCO instance.

All of this suggests that computational complexity may not be a very strong barrier against manipulation. Therefore, the next step is to investigate other approaches to prevent manipulation. In this dissertation I will discuss two promising ideas. In Chapter 6, we show that restricting the manipulators' information about the other voters can make a natural type of manipulation (which we call *dominating manipulation*) computationally hard, or even make such manipulations impossible.[6] In Chapter 12 we aim at obtaining and characterizing strategy-proof voting rules for combinatorial voting, by restricting the voters' preferences, which, as we discussed in the introduction, is a method that has traditionally been pursued by economists.[7]

3.2 Game Theory and Voting

Game Theory is a useful tool to model strategic situations (for an overview, see (Fudenberg and Tirole, 1991)). Game Theory has been extensively used in many disciplines, including Economics, Political Science, Computer Science, Statistics, and even Biology. In particular, Game Theory is often used in Multi-Agent Systems (Shoham and Leyton-Brown, 2009). The most basic type of games, called a *normal-form game*, consists of the following parts.

1. There is a finite set of n players (agents).
2. For each agent i, there is a finite set of actions A_i. A vector in $A_1 \times \cdots \times A_n$ is called an *action profile*.
3. For each agent i, there is a real-valued utility function u_i that maps each action profile to a real number. This utility function models the agent's preferences

[6] Published in Conitzer et al. (2011a).

[7] Published in Xia and Conitzer (2010c).

over all action profiles.

Example 3.2.1 (Prisoner's dilemma)**.** *There are two players (prisoners) who can choose to either cooperate (***C***) with each other or defect (***D***). If both of them cooperate, then both will stay in prison for one month; if both of them defect, then both will stay in prison for five months; if one cooperate and the other defect, then the player who cooperates will stay in prison for 10 months, and the player who defects will be released immediately. The utility functions of the players are depicted in Table 3.2.*

Table 3.2: The prisoner's dilemma.

	C	**D**
C	(-1,-1)	(-10,0)
D	(0,-10)	(-5,-5)

In a vector (a, b) *in the table,* a *is the utility of the row player and* b *is the utility of the column player. In this game we model a player's utility by the negation of the number of months he will be imprisoned.*

Having set up the game, we can predict the outcome of the game by investigating some solution concepts. *(Pure) Nash Equilibrium (NE)* is one of the most famous solution concepts. A pure Nash equilibrium is defined to be an action profile where no player can benefit from deviating to another action, assuming that all of the other players do not change their actions. For example, the only pure NE of the game in Example 3.2.1 is the action profile where both players defect. When both of them defect, the row player has no incentive to change his action to cooperate, because this would only lower his utility from -5 to -10. Similarly, in this case the column player also has no incentive to deviate. Therefore, the action profile where both players defect is an NE. To show that this is the only NE of the game, we observe that (1) if both of them cooperate, then either player has an incentive to change his action to **D**, because this will raise his utility from -1 to 0, and (2) if one player cooperates and the other player defects, then the former has an incentive to change his action to **D**, because this will raise his utility from -10 to -5. Hence, there is no other NE except the action profile where both players defect.

In a game, if for a player the following two conditions hold: (1) choosing an action a never gives her a lower utility than choosing another action b, no matter what the other players' action are, and (2) sometimes choosing a gives her a strictly higher utility than choosing b, then we say that *a (weakly) dominates b* for that player. Here b is said to be *(weakly) dominated*. If choosing a always gives the player a strictly higher utility than choosing b, then we say that *a strictly dominates b*. For example, in the prisoner's dilemma (Example 3.2.1), **C** is strictly dominated by **D**. A strictly dominated action will never be played in an NE.

In a voting setting we use linear orders over alternatives to model voters preferences, instead of utility functions. Hence, the simultaneous-move voting games are defined as follows.

Definition 3.2.2. *A simultaneous-move voting game consists of the following components.*

- *There is a set of m alternatives $\mathcal{C}$ and a set of n voters (players).*
- *For each voter, the set of actions is $L(\mathcal{C})$, which is the set of all linear orders over $\mathcal{C}$.*
- *There is a voting rule r that selects a unique winner for each profile.*
- *For each voter, there is a linear order over $\mathcal{C}$ that represents her true preferences.*

The concept of pure NE naturally carries over to simultaneous-move voting games. In such games, a pure NE is a profile where no single voter can improve the winner by casting a different vote, assuming that the other voters do not change their votes. Unlike in the prisoner's dilemma, where there is only one NE, for almost all common voting rules there are many trivial NE. In fact, for all common voting rules, if the number of voters is large enough, then there are many profiles where no single voter can even change the winner by voting differently. For example, suppose there are three voters whose true preferences are Obama≻Clinton≻McCain, and the plurality rule with lexicographic tie-breaking is used to select the winner. In the profile where all three voters vote for McCain≻Clinton≻Obama, no single voter can change the winner by voting differently. Therefore, this profile is an NE, in which the winner is the least preferred alternative in all voters' true preferences. It is easy to see that, generally, any alternative is the winner in some NE of simultaneous-move voting games. This observation suggests that pure Nash equilibrium, as a solution concept, is too coarse for analyzing voting games. One refinement was proposed by Farquharson (1969), who proposed to focus on Nash equilibria in a reduced voting game, where all iteratively dominated votes are eliminated. However, after iteratively removing all dominated votes, in general there still too many Nash equilibria.

A solution concept that will play an important role in this dissertation is *subgame-perfect Nash Equilibrium (SPNE)*. SPNE are defined for *extensive-form games*, which consist of multiple stages. For simplicity, here we only define extensive-form games with perfect information.

Definition 3.2.3. *An extensive-form game with perfect information is represented by a tree and the following components.*

- *Each (decision) vertex of the tree is labeled by a player, who chooses an action at the vertex. Each action corresponds to an edge going deeper towards the leaves.*

- *Each leaf node is associated with an outcome vector that assigns a real value to* *each player.*

Example 3.2.4. *Consider the two prisoners in Example 3.2.1. Now suppose that in the first stage the row player (player* 1*) chooses to cooperate or defect. Then in the second stage, the column player (player* 2*) chooses his action. Furthermore, we suppose that player* 2 *can observe player* 1*'s action (that is, he has perfect information about player* 1*'s move). This situation can be modeled by the extensive-form game depicted in Figure 3.1.*

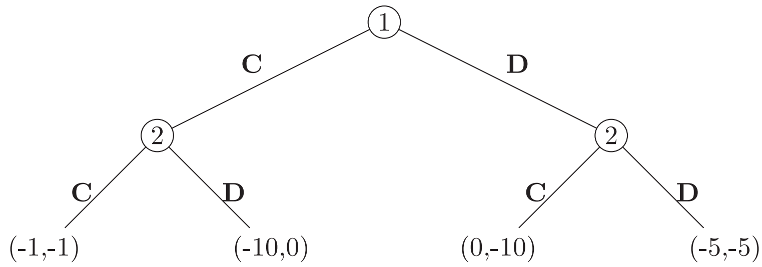

FIGURE 3.1: An extensive-form game.

In extensive-form games, a player must choose an action for each of her decision vertices. All these actions together constitute a *strategy* of the player. A *strategy profile* constitutes a strategy for each player. A (pure) subgame-perfect Nash equilibrium is not only a Nash equilibrium, but is also a Nash equilibrium of the extensive-form game represented by any sub-tree of the original extensive-form game. For example, the only SPNE of the game in Example 3.2.4 is the strategy profile where player 1 chooses to defect, and player 2 chooses to defect at both of his decision vertices. SPNE can be computed by a technique called *backward induction*, which starts with the bottom decision vertices, and computes the optimal actions for the players at these decision vertices. Then, we move up to the layer above it. Since we can predict the outcome at each decision vertex in the lower layer, we can then compute the optimal actions for the players at the decision vertices in the current layer. And after this, we move up to the next layer above, etc., until the optimal action for the root vertex has been computed.

It is straightforward to define an extensive-form game for voting. In Chapter 7, we will study an extensive-form voting game where voters vote one after another. We call such games *Stackelberg voting games*. One nice property about Stackelberg voting games is that, for each Stackelberg voting game, the winner is the same in all SPNE. This allows us to focus on analyzing the quality of the winner, rather than analyzing which NE should be the outcome (this problem is known as the *equilibrium selection problem*). In Chapter 11, we will see another extensive-form game defined specifically for combinatorial voting, where the voters vote simultaneously, but they vote over one issue after another. We call such games *strategic sequential voting processes (SSP)*. For SSP we will focus on a solution concept that is similar to SPNE. Under this solution concept, the winner in any such SSP is unique, and can be computed by a technique that is similar to backward induction.

3.3 Summary

In this chapter, we reviewed some literature on game-theoretic aspects of voting. In Section 3.1, I gave a brief overview of previous work on using computational complexity as a barrier against manipulation. In Section 3.2, we recalled basic definitions of normal-form and extensive-form games, and solution concepts such as (pure) Nash equilibrium and (pure) subgame-perfect Nash equilibrium. We also pointed out that the biggest challenge in analyzing simultaneous-move voting games is the equilibrium selection problem, which, as we will see, is alleviated in extensive-form voting games.